Copy 37 of 123 copies

2

IDA PAPER P-2248

COMBAT: A COMPUTER PROGRAM TO INVESTIGATE AIMED FIRE ATTRITION EQUATIONS, ALLOCATIONS OF FIRE, AND THE CALCULATION OF WEAPONS SCORES

Lowell Bruce Anderson
Frederic A. Miercort

September 1989

IDA
INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311-1772

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, or (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevence to the problems studied, and they are released by the President of IDA.

Group Reports

Group Reports record the findings and results of IDA established working groups and panels composed of senior individuals addressing major issues which otherwise would be the subject of an IDA Report. IDA Group Reports are reviewed by the senior individuals responsible for the project and others as selected by IDA to ensure their high quality and relevance to the problems studied, and are released by the President of IDA.

Papers

Papers, also authoritative and carefully considered products of IDA, address studies that are narrower in scope than those covered in Reports. IDA Papers are reviewed to ensure that they meet the high standards expected of refereed papers in professional journals or formal Agency reports.

Documents

IDA Documents are used for the convenience of the sponsors or the analysts (a) to record substantive work done in quick reaction studies, (b) to record the proceedings of conferences and meetings, (c) to make available preliminary and tentative results of analyses, (d) to record data developed in the course of an investigation, or (e) to forward information that is essentially unanalyzed and unevaluated. The review of IDA Documents is suited to their content and intended use.

The work reported in this publication was conducted under IDA's Independent Research Program. Its publication does not imply endorsement by the Department of Defense or any other Government Agency, nor should the contents be construed as reflecting the official position of any Government Agency.

This Paper has been reviewed by IDA to assure that it meets the high standards of thoroughness, objectivity, and appropriate analytical methodology and that the results, conclusions and recommendations are properly supported by the material presented.

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***

2. REPORT DATE
September 1989

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
COMBAT: A Computer Program to Investigate Aimed Fire Attrition Equations, Allocations of Fire, and the Calculation of Weapons Scores

5. FUNDING NUMBERS
Independent Research

6. AUTHOR(S)
Lowell Bruce Anderson, Frederic A. Miercort

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-2248

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

This paper describes and summarily documents a computer program called COMBAT, and it discusses in detail the methodological concepts behind that program. COMBAT is designed to investigate selected subsets of three aspects of the modeling of combat. The aspects considered are: formulas used to compute attrition in combat models, formulas used to compute allocations of fire in combat models, and formulas used to compute relative measures of force effectiveness that result from combat models.

Three potential uses for the COMBAT computer program are as follows. First, it can be used as a research tool to investigate the characteristics and interrelationships of various formulas that compute attrition, allocation of fire, and force effectiveness measures. Second, code can be extracted from the COMBAT computer program (or this code could serve as a prototype) for use in other (more detailed) models of combat. Third, COMBAT can be used as a highly aggregated, stand-alone model of conventional combat. A copy of the code of COMBAT on a 5.25-inch disk (PC/MS-DOS format) is attached to the inside base cover of the paper. If this disk is missing, another coppy can be obtained from the authors at the Institute for Defense Analyses.

14. SUBJECT TERMS
Lanchester Equations, Antipotential Potential, Eigenvector Scores for Weapons Systems, Measures of Effectiveness for Comparing Military Forces, Coordination of Fire, Shoot-look-shoot fire, Deterministic Modeling of Attrition, Dynamic Simulation of Warfare

15. NUMBER OF PAGES
202

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

IDA PAPER P-2248

COMBAT: A COMPUTER PROGRAM TO INVESTIGATE AIMED FIRE ATTRITION EQUATIONS, ALLOCATIONS OF FIRE, AND THE CALCULATION OF WEAPONS SCORES

Lowell Bruce Anderson
Frederic A. Miercort

September 1989

INSTITUTE FOR DEFENSE ANALYSES

IDA Independent Research Program

PREFACE

This paper was prepared as part of IDA Project 9000-109 under the IDA Independent Research Program.

This paper describes a computer program designed to investigate three aspects of the modeling of combat. These aspects are: formulas used to compute attrition, formulas used to compute allocations of fire, and formulas used to compute relative measures of force effectiveness. The methodological concepts underlying these formulas are discussed in some detail.

The authors are grateful to Dr. Alan Rolfe and Dr. Leo Schmidt for their helpful reviews of this paper.

ABSTRACT

This paper describes and summarily documents a computer program called COMBAT, and it discusses in detail the methodological concepts behind that program. While COMBAT can be viewed as being a general model of conventional combat, it was not primarily designed for that purpose. Instead, COMBAT was designed to investigate selected subsets of three aspects of the modeling of combat. The three aspects considered are: formulas used to compute attrition in combat models, formulas used to compute allocations of fire in combat models, and formulas used to compute relative measures of force effectiveness that result from combat models.

Three potential uses for the COMBAT computer program are as follows. First, it can be used as a research tool to investigate the characteristics and interrelationships of various formulas that can be used to compute attrition, allocation of fire, and force effectiveness measures. Second, code can be extracted from the COMBAT computer program (or this code could serve as a prototype) for use in other (more detailed) models of combat. Third, as stated above, COMBAT can be used as a highly aggregated, general purpose model of conventional combat. Some limitations concerning this use of COMBAT are discussed in the paper.

COMBAT is programmed in FORTRAN-77. A copy of its code on a 5.25-inch disk (PC/MS-DOS format) has been attached to the inside back cover of this paper. If this disk is missing, another copy can be obtained from the authors at the Institute for Defense Analyses.

CONTENTS

Preface iii

Abstract v

I. Introduction I-1

II. Weapons Scores and Force Comparisons II-1

A. General Structure II-1

B. Notation II-3

C. Alternative Methods for Calculating W:apons Scores II-6

1. Antipotential Potential II-6

2. Antipotential Potential with Vulnerability II-7

3. Potential Exchange Potential II-11

4. Lethality/Vulnerability Potential II-12

5. Dynamic Potential II-14

6. Rescaling II-16

D. Aircraft II-17

1. The Consideration of Combat Aircraft as Fully Interacting Weapons Systems II-18

2. The Consideration of Ground Attack Aircraft as Non-Vulnerable Weapons Systems II-18

a. Additional Notation for Non-Vulnerable Weapons Systems II-19

b. The Calculation of Lethality Scores for Non-Vulnerable Weapons Systems II-20

E. Alternative Measures for Compacing Forces II-22

1. Background II-22

2. Structure II-23

3. Means II-24

4. Measures II-26

5. References II-27

F. Setting Weapons Scores by Direct Input II-29

III. Allocations of Fire that do not Depend on Weapons Scores III-1

A. Introduction III-1

1. Background III-1

2. Numbering These Allocation Methods III-6

B. Method 0: Allocations that are Proportional to Weighted Numbers of Targets III-7

C. Method 1: Al'ocations that are Proportional to Probabilities of Kill Times Weighted Numbers of Targets III-10

D. Method 2: Allocations that are a Convex Combinatior of a Priority Based on Pk's and Method 1 .... III-12

E. Method 3: Allocations that are a Convex Combination of a Priority Based on Pk's and Method 0 .... III-15

F. Method 4: Allocations that are Determined by Group Detections and Rigid Priorities Based on Pk's III-16

IV. Allocations of Fire that Depend on Weapons Scores IV-1

A. Structure IV-1

B. Methods to Allocate Fire that Depend on Weapons Scores IV-3

V. Attrition Equations for Aimed Fire V-1

A. Structure V-1

1. General Approaches for Modeling Attrition V-1

2. The Deterministic Estimation Approach for Modeling Attrition V-2

3. General Approaches for Processing Time V-3

4. The Time-Step Approach with Relatively Long Time Periods for Processing Time V-4

5. Implications and Resulting Structure V-6

B. Unilateral Attrition Equations V-7

1. Notation for Unilateral Attrition V-8

2. Overview of Options for Calculating Unilateral Attrition V-10

3. Attrition Assuming Uncoordinated Fire V-11

a. Assumptions V-11

b. Resulting Attrition Equation V-12

4. Attrition Assuming Uniformly Coordinated Fire V-13

a. Assumptions for Homogeneous Shooters and Heterogeneous Targets V-13

b. Resulting Attrition Equation for Homogeneous Shooters and Heterogeneous Targets V-15

c. Assumptions for Heterogeneous Shooters and Targets V-15

d. A Consistent Attrition Equation for Heterogeneous Shooters and Targets V-16

5. Attrition Assuming a Mix of Uncoordinated and Uniformly Coordinated Fire V-17

a. Assumptions V-17

b. A Consistent Attrition Equation V-17

6. Attrition Assuming a Shoot-Look-Shoot Firing Process V-18

a. Assumptions for Heterogeneous Shooters and Homogeneous Targets V-18

b. Resulting Attrition Process for Heterogeneous Shooters and Homogeneous Targets V-19

c. A Simple Extension to Consider Heterogeneous Shooters and Targets V-20

d. Comments V-22

7. Attrition Assuming a Lanchester Square Process V-22

a. Assumptions V-22

b. Resulting Attrition Equation V-25

8. Summary V-26

a. Relationships Among These Attrition Equations V-26

b. Other Sets of Assumptions V-28

C. Converting Unilateral Attrition Assessments into Bilateral Attrition V-29

1. Notation and Equations V-30

2. Discussion V-31

VI. On Using COMBAT as a Dynamic Simulation of Warfare VI-1

A. Limitations that Affect Both Ground and Air Combat VI-1

1. Resources VI-1

a. Replacements VI-1

b. Munitions VI-2

c. People and Supplies VI-2

2. Interactions VI-2

a. Battle Damage and Repair VI-2

b. Non-Battle Damage and Destruction VI-2

c. Effectiveness Parameters that are Functions of Attack and Defense VI-3

d. Degrading the Nominal Effectiveness of Unbalanced Forces VI-3

e. Area Fire VI-3

3. Geography VI-3

a. Geography and Attrition VI-3

b. Depth from the Front VI-3

c. Maneuver VI-4

d. Sectors VI-4

4. Output Measures VI-6

a. Killer-Victim Scoreboards VI-6

b. FEBA Movement VI-6

c. Different Uses for Different Scores VI-6

5. Changing Data Values During a Simulation VI-7

6. User Interface VI-7

B. Limitations that Primarily Affect Air Combat VI-7

1. Fixed Assignments of Aircraft to Missions VI-8

2. Non-Lethal Suppression VI-8

3. Timing of Interactions VI-9

4. Selected Characteristics of Deep Strikes VI-9

a. Interdiction Missions VI-9

b. Air Base Attack Missions VI-9

5. Summary VI-10

VII. Summary Documentation of the COMBAT Computer Program VII-1

A. Inputs VII-1

1. Input Data VII-1

2. Input Files and Procedures VII-9

B. Symbolic Constants VII-20

C. Outputs VII-20

1. Output Files VII-20

2. Output Data VII-21

a. Output of Inputs VII-21

b. Some Calculated Arrays VII-21

c. Major Results Displayed on the ".OUT" File VII-24

d. Major Results Displayed on the ".SUM" File VII-26

D. Source Code VII-27

E. Running COMBAT VII-33

References R-1

Appendix A. An Introduction to Antipotential Potential Weapon Scores A-1

Appendix B. A Robust Method for Considering Effectiveness Parameters that are Functions of Attack and Defense B-1

Appendix C. An Improved Method for Degrading the Nominal Effectiveness of Unbalanced Forces C-1

Appendix D. A General Method for Relating Bounds on Attrition to the Average Rate of Movement of Ground Forces D-1

Appendix E. A Generalized Method for Calculating Force Strengths E-1

List of Tables

IV-1. A Su⁻ mary of Selected Characteristics Concerning Five General Approaches for Relating Allocations of Fire and Weapons Scores........... IV-4

VII-1. Relationship Between the FORTRAN Inputs to COMBAT and the Algebraic Notation of Chapters II through V.................................. VII-2

VII-2. Definitions of FILEIN, IALMTH, IALVAL, IALWT, IALWTA, IATRTE, IFVUL, INVAL, IRSCRN, IVSUMP, IWSCRN, IWSCRO, NAMVAR, NPER, and NWNAME VII-5

VII-3 Definitions of IALMTH, IALVAL, and INVAL.............................. VII-8

VII-4. Range of Allowable Values of Inputs..VII-10

VII-5. Special Restrictions that Apply to IALVAL(IS).............................VII-11

VII-6. File Inputs, In Order, if IRSCRN = 0 ....VII-12

VII-7. Keyboard Inputs (In the Order in Which They Can be Requested) if IRSCRN = 1...VII-16

VII-8. File Inputs, In Order, if IRSCRN = 1 ....VII-17

VII-9. Calculated Arrays Written to Output Files....................................VII-22

VII-10. COMBAT Source Code Files..VII-28

VII-11. The Total Size of COMBAT and of Each of Its Program Units.............VII-30

List of Figures

VII-1. A Sample Input Data File with IRSCRN = 0.................................VII-15

VII-2. A Sample Input Data File with IRSCRN = 1.................................VII-19

VII-3. A Calling Tree of Subroutines and Functions for Program COMBAT.....VII-29

I. INTRODUCTION

The purpose of this paper is to describe and summarily document a computer program called COMBAT, and to discuss in detail the methodological concepts behind that program. While COMBAT can be viewed as being a general model of conventional combat, it was not primarily designed for that purpose. Nor does COMBAT model specific combat missions or battles. Instead, COMBAT allows the investigation of selected subsets of three aspects of the modeling of combat. This restriction to "subsets of aspects of the modeling of combat" is somewhat confining, and COMBAT is a relatively small computer program. However, the topics discussed here are, in a reasonable sense, major subsets of important aspects of a significant class of combat models.

The class of combat models considered here are those that group the weapons systems on each side into one or more types of weapons, and then assess attrition by considering the number of weapons of each type on each of two sides, along with a variety of effectiveness parameters, to yield the number of weapons of each type destroyed on each side. This is one of the two most frequently used approaches for modeling combat.[1] This "aggregation into types" approach is useful both for constructing large-scale combat models (such as theater or campaign-level models) and for constructing quick-running combat models. This paper does not analyze the relative strengths and limitations of this approach as compared to other approaches (such as the Monte-Carlo approach). Instead, it addresses three aspects of this approach given that this general approach is being considered.

The three aspects of this approach considered here are: the formulas used to compute attrition, the formulas used to compute allocations of fire, and the formulas used to compute relative measures of force effectiveness. These aspects are clearly related to each other, but the degree of their interrelationships can vary. For example, if all of the weapons systems on each side are grouped into one generic type of weapons system, then

[1] The other common approach keeps track of each individual weapons system. In that approach, weapons systems are individually considered, in part by comparing randomly-drawn numbers to input probability distributions (i.e., using Monte Carlo techniques) in order to determine the occurrence of various events involving each weapons system, such as the attrition to those weapons systems involved in a particular engagement.

allocation of fire plays no role, yet the choice of formulas to compute attrition and to measure relative force effectiveness remain important. Conversely, if the weapons systems are grouped into more than one type, then attrition will depend on the allocation of fire, and the allocation of fire will (in general) depend on previous attrition. Further, computation of force measures will (in general) depend on the capabilities of the weapons system being considered which can include how their fire is allocated to the various types of enemy targets, and the method of allocating this fire can depend (in part) on how the capabilities of the enemy weapons systems are measured. The COMBAT computer program can be used to examine selected characteristics of each of these aspects of the modeling of combat individually, as well as examining selected characteristics of their interrelationships.

Three potential uses for the COMBAT computer program are as follows. First, it can be used as a research tool to investigate the characteristics and interrelationships of various formulas that can be used to compute attrition, allocation of fire, and force effectiveness measures. Second, code can be extracted from the COMBAT computer program (or this code could serve as a prototype) for use in other (more detailed) models of combat. Third, as stated above, COMBAT can be used as a highly aggregated, general purpose model of conventional combat. (Some limitations concerning this use of COMBAT are discussed in Chapter VI, below.)

The remainder of this paper is organized as follows. Chapter II discusses various methods for calculating weapons scores and force comparisons based on these scores. When there is more than one type of weapon on (at least one of) the two sides, these methods all depend, in part, on the allocation of fire of each type of weapon. For the purposes of Chapter II, it is assumed that these allocations of fire are known.

Chapter III discusses various methods for computing allocations of fire that are independent of weapons scores. Accordingly, one set of options available in COMBAT is found by combining these options for calculating allocations of fire (which do not depend on weapons scores) with the options in Chapter II for computing weapons scores (given allocations of fire).

Chapter IV discusses various methods for computing allocations of fire that depend on weapons scores. Since weapons scores depend on allocations of fire, these methods yield sets of simultaneous equations which must then be solved to obtain both the allocations of fire and the associated weapons scores. (Note that the methods described in

Chapter III require only evaluating formulas to determine allocations of fire, not solving simultaneous equations.)

Chapter V discusses various methods for computing attrition given that allocations of fire are known. These methods all concern point fire; they differ in the degree of coordination of fire. The incorporation of these attrition methodologies into the computer program means that COMBAT can (optionally) make use of the allocations of fire computed as described in Chapters III or IV in these attrition methodologies to yield a dynamic model of conventional combat. Chapter VI discusses some limitations concerning this use of COMBAT, and it suggests a few alternatives to mitigate some of these limitations. Of course, the code of COMBAT could be expanded, if desired, to directly address these limitations.

Chapter VII presents a summary documentation of the COMBAT computer program. In this documentation, all of the input variables and arrays for COMBAT are defined (either directly or by reference to the appropriate parts of previous chapters), the ranges of allowable values for these inputs are presented, and the procedures for entering these inputs are discussed. Symbolic constants that are used as dimension bounds in array declarators are defined and described. The output files produced by COMBAT are discussed. The files that contain the source code for COMBAT are listed, as are its program units, and a calling tree of subroutines and functions is given. Finally, the sizes of the various program units that comprise COMBAT, as well as its total size, are stated. This summary documentation does not contain an exhaustive set of definitions of all of the major working variables and arrays (many, but not all, are given; see Chapter VII for details). Descriptions of the internal operations of COMBAT's program units are not given, nor are estimates of its computer running time (which depends heavily on the speed of the computer being used, on the numbers of types of weapons being considered, and (in some cases) on the methodological options being selected).

COMBAT is programmed in FORTRAN-77. A copy of its code on a 5.25-inch disk (PC/MS-DOS format) has been attached to the inside back cover of this paper. If this disk is missing, another copy can be obtained from the authors at the Institute for Defense Analyses.

II. WEAPONS SCORES AND FORCE COMPARISONS

A. GENERAL STRUCTURE

Quantitative comparisons of the forces on two opposing sides are frequently formed in the following fashion. First, all of the resources on each of the two sides are grouped into a set of categories. Each of the resources in each category is assigned a (non-negative) value or score, where these scores are constant within categories, can vary across categories, and can be functions of the numbers and effectiveness parameters of the resources on both sides. While these categories need not be the same for the two sides, each side will have at least one category consisting of resources that (perhaps implicitly) has been assigned a score of zero. Force strengths are then formed for each side by summing over all categories the product of the number of resources in that category times the score given to the resources in that category. A quantitative comparison of the two forces is then calculated as a (real-valued) function of these force strengths and, perhaps, of exogenous input parameters. In particular, such force comparison functions depend on the number and effectiveness of the weapons in the forces only through these force strengths.

The most common such force comparison measure is simply the ratio of force strengths, i.e., the force ratio based on these force strengths. Another force comparison measure is the estimated average distance that the forces would move in some time period, which is frequently calculated as a non-linear function of force ratio and other parameters (such as posture and terrain). Many other force comparison measures are possible and, as discussed below, several others are available in COMBAT.

Accordingly, if a quantitative comparison of the forces on two opposing sides is desired, then it is reasonable to consider the following questions. First, should the structure posed above be followed, or should some other structure be used? Second, if this structure is to be followed, how should the resources be partitioned into categories? In particular, which resources should be put into the category that is, by assumption, assigned a score of zero (e.g., which are not going to be explicitly addressed), and how should the resources to be explicitly addressed be grouped into types? Third, how should the scores for the resources in each type be calculated? Finally, how should the resulting force strengths be combined to form a comparative measure of the effectiveness of these forces?

Obtaining definitive answers to all of these questions is beyond the scope of this paper.

A quick but useful answer to the first question is as follows. If at all possible, this structure should be used only in conjunction with a dynamic model of combat. In the past, dynamic models of combat tended to be very complex, and took a long time to run on large and relatively inaccessible computers. Hence, there was a practical need for quantitative alternatives to evaluating forces using dynamic models. Now, however, dynamic combat models that range from being relatively simple to being somewhat complex can be run very quickly on very accessible personal computers (as well as on computers accessible through terminals). Thus, important aspects concerning the dynamics of combat, many of which simply cannot be adequately addressed using only numbers of weapons and weapons scores, can now be relatively easily addressed using dynamic models. Weapons scores and force comparison measures remain important, but in a different role; namely, to help assess the results of dynamic models.

A quick answer to the second question is as follows. As always, there are (in general) too many variations in resources to group only those resources that are absolutely identical to each other into the same type, and to consider all resulting possible types. However, modern computers allow the practical consideration of more types than previously could be addressed, and changes in these types can be made more easily and quickly. Accordingly, decisions here will still be required, and these decisions will still be somewhat hard to make, but they can be made at a more fine-grained level and hence be relatively less important than before. That is, given the capabilities of modern computers, it seems better to initially err on the side of considering too many different types of resources rather than too few, and then (perhaps) to cut down on the number of types being considered over the course of a study, where some resources may be more finely grouped into types than others, depending on the issues being studied.

This paper presents several alternative answers to the third question (how should scores be calculated) and fourth question (how should force comparisons be addressed) posed above. However, this paper does not recommend any one particular method over the others. As indicated above, COMBAT can be used as a tool for investigating the properties and suitability of each of these alternative methods.

B. NOTATION

Given the general structure posed above, suppose that it has been decided to group the resources on each of two sid‹s into one of three general categories: those resources not being explicitly considered,[1] weapons systems whose full lethality and vulnerability are to be directly addressed, and (optionally) weapons systems whose lethality is to be considered but whose vulnerability is not to be addressed.

The reason for this third category of weapons systems is as follows. Several dynamic models use scores for ground weapons in ground-to-ground combat and for aircraft (on close air support missions) attacking enemy ground weapons, but not for any other missions for aircraft. In these models, all relevant interactions for ground weapons are fully considered in their ground combat portion, but only the lethality of aircraft against ground weapons is considered there. The vulnerability of these aircraft to enemy air base attacks, enemy defensive aircraft, and enemy surface-to-air missiles (SAMs) and antiaircraft artillery (AAA) is considered separately in the air combat portion of these models. Accordingly, for these (and similar) models, scores are needed for systems in both of the latter two categories described just above.

COMBAT assumes that there is at least one type of weapon in the second category (full consideration of lethality and vulnerability), but there need not be any in the third category. The discussion below initially assumes that all of the weapons systems being considered belong to this second category. Later in this chapter, additional notation and calculations concerning systems in the third category (if any) will be discussed.

Let the two sides be denoted by side 1 and side 2, and assume that all of the (fully interacting) weapons systems being considered on side s have been partitioned into N^s types, where $s = 1{,}2$ and $N^s > 0$ for both s. Consider the following notation:

W_i^s = the number of weapons of type i on side s, where $i = 1,N^s$ and $s = 1,2$.[2]

[1] For the purpose of this paper, this category will include all resources that are not weapons systems capable of inflicting attrition on the enemy.

[2] The notation $i = 1,I$ will be used interchangeably with $i = 1,\ldots,I$ throughout this paper.

E_i^s = the average number of engagements per time period (or per unit time) made by a weapon of type i on side s (against all enemy weapons), where i = 1,N^s and s = 1,2.

P_{ij}^s = the probability of kill per engagement by a weapon of type i on side s when that weapon is engaging an enemy weapon of type j, where i = 1,N^s, j = 1,$N^{s'}$, s = 1,2, and s' = 3–s.

$\tilde{A}^a$: For each allocation method a (the range of a is described in Chapter III, Section A.2), $\tilde{A}^a$ is a set of input parameters used in conjunction with W, E, and P to calculate allocations of fire. (Specifications for $\tilde{A}^a$ will be discussed in Chapters III and IV.)

F_{ij}^{as}: For each allocation method a, F_{ij}^{as} is a function that maps W, E, P, and $\tilde{A}^a$ into the average fraction of engagements (i.e., the allocation of fire) that weapons of type i on side s make against enemy weapons of type j (out of all of the engagements made by those type-i weapons), where i = 1,N^s, j = 1,$N^{s'}$, s = 1,2, and s' = 3–s. (Various formulas for F^a will be discussed in Chapters III and IV.)

A_{ij}^s = $A_{ij}^s(a) = F_{ij}^{as}(W,E,P,\tilde{A}^a)$, where i = 1,$N^s$, j = $N^{s'}$, s = 1,2, and s' = 3–s.

K_{ij}^s = $E_i^s A_{ij}^s P_{ij}^s$, where i = 1,N^s, j = $N^{s'}$, s = 1,2, and s' = 3–s. (Thus, K_{ij}^s gives a measure of the capability of an average weapon of type i on side s to kill enemy weapons of type j.)

h = the index of the type of weapon on side 1 whose score will be set to 1.0 (i.e., h is used to scale weapons scores), where h ∈ $\{1,\ldots,N^1\}$.

G_i^{vs}: For each scoring method v (the range of v will be discussed below), G_i^{vs} is a function that maps W, K, and h into a (static) score for weapons of type i on side s, where i = 1,N^s and s = 1,2.

$V_i^s(v) = G_i^{vs}(W,K,h)$, where $i = 1,N^s$ and $s = 1,2$. Thus, for each scoring method v, $V_i^s(v)$ is the score for weapons of type i on side s.

As indicated above, Chapters III and IV consider alternative (explicit) specifications for the function F. This chapter considers alternative implicit specifications for the function G. These specifications are implicit in that, for each v other than v = 2, for each i from 1 to N^s, and for both s, $V_i^s(v)$ is defined in terms of W, K, h, and V(v). Thus, this structure results in at least $N^1 + N^2$ equations with at least $N^1 + N^2$ unknowns, where the given parameters are denoted by W, K, and h, and $N^1 + N^2$ of the unknowns are denoted by:

$$V_1^1(v),\ldots,V_{N^1}^1(v), V_1^2(v),\ldots,V_{N^2}^2(v) .$$

(An essentially equivalent comment applies when v = 2.) If the matrix given by the matrix product of K^1 times K^2 is irreducible, then a unique solution for V(1) is known to exist. Solutions may or may not exist and (if they exist) may or may not be unique for the other values of v. Subject to numerical stability, COMBAT will find a solution for the v = 1 case, which will be the unique solution if the aforementioned product matrix is irreducible. For the other cases, COMBAT attempts to iterate to some solution, and it will display warning messages if no solution is found. COMBAT does not address whether any solution it finds is unique or not.

Note that V is defined in terms of W, K, and h, and that the notation above gives one particular definition for K. It is important to note that several other reasonable definitions for K exist. For example, K could be defined to be a different function of E, A, and P (and, perhaps, of other parameters), and to be a direct function of W (instead of only implicitly depending on W through A). Such alter. .ive functions could be obtained by taking partial derivatives with respect to the number of shooting weapons (of type i on side s) of an attrition equation that calculates the number of target weapons (of type j on side s') killed as a function of W, E, A, P, and then evaluating those partial derivatives at the force sizes in question. Using different attrition equations would yield different definitions for K; several different attrition equations that could be considered are presented in Chapter V below. COMBAT has not (yet) been coded to allow investigation of alternative definitions for K--the current COMBAT computer program allows only the definition for K indicated in the notation above. Future research could examine the impact of alternative definitions for K. See Reference [29] for a discussion of some initial work in this area. The remainder of this chapter assumes only that K has been specified in some manner, and it discusses alternative functional forms for mapping {W,K,h} to V.

C. ALTERNATIVE METHODS FOR CALCULATING WEAPONS SCORES

All but one of the scoring methods described below (i.e., all but the antipotential potential method discussed in Section 1) make use of the quantity $W_i^s K_{ij}^s / W_j^{s'}$ when $W_j^{s'} > 0$. Accordingly, it is useful to define $\hat{K}$ as

$$\hat{K}_{ij}^s = \begin{cases} W_i^s K_{ij}^s / W_j^{s'} & W_j^{s'} > 0 \\ 0 & \text{otherwise}, \end{cases}$$

for i = 1,N^s, j = 1,$N^{s'}$, s = 1,2, and s' = 3–s. Just as K_{ij}^s gives a rate at which one weapon of type i on side s is killing (all) type-j weapons on side s', $\hat{K}_{ij}^s$ gives a rate at which each type-j weapon on side s' is being killed by (all) weapons of type i on side s.

1. Antipotential Potential

The antipotential potential (APP) method for calculating weapons scores assumes that these scores (denoted by $V_i^s(1)$ for i = 1 to N^s and s = 1 and 2) must satisfy the general relationship that

$$\beta V_i^s(1) = \sum_{j=1}^{N^{s'}} K_{ij}^s V_j^{s'}(1) \qquad i = 1,N^S,\ s = 1,2,\ s' = 3-s,$$

for some β (β is calculated along with the weapons scores), and that $V_h^1(1)$ must satisfy the scaling relationship that

$$V_h^1(1) = 1.$$

The basic idea behind this general relationship is that the value of a type-i weapon on side s, $V_i^s(1)$, should be proportional to the sum over j of the rate at which that weapon is killing enemy weapons of type j, K^{s}_{ij}, times the value of a type-j enemy weapon, $V^{s'}_{j}(1)$, where the proportionality factor, β, is independent of weapon type and side.

Antipotential potential is discussed further in Appendix A (see also the references of that appendix and Section B of Reference [27]), and is used in various ways in IDAGAM, INBATIM, TACWAR, JCS FPM, and IDAPLAN, all of which are dynamic theater-level models of ground and air combat.

One characteristic of APP is that if two different types of weapons systems have identical capabilities to kill enemy weapons then they will receive the same APP score, no matter whether one is more vulnerable than the other. This characteristic may not be a meaningful flaw when APP is used inside of a dynamic model, but it could be a serious flaw if APP is to be used as a stand-alone measure. The other four methods described below for calculating weapons scores were developed (in part) to alleviate this potential flaw, and (in particular) none of them have this characteristic. As indicated above, all four of these alternative methods below make use of the array $\hat{K}$, which APP does not use. Accordingly, APP is the only method for calculating weapon scores presented here that, given K, does not depend on W. (Note, however, that K depends on A which, in general, depends on W.)

2. Antipotential Potential with Vulnerability

The antipotential potential with vulnerability (APPVUL) method is, in some senses, the closest of these four other methods to APP. One sense in which it is close to APP is that the scaling assumption used in APPVUL requires that the force ratio produced by APPVUL be equal to that produced by APP. That is, for i = 1,N^s and s = 1,2, let $V^{s}_{i}(1)$ give the weapons scores produced by APP as described above, and set

$$R(1) = \frac{\sum_{j=1}^{N^2} W^2_j V^2_j(1)}{\sum_{i=1}^{N^1} W^1_i V^i_i(1)} .$$

For i = 1,N^s and s = 1,2, let $V_i^s(2)$ give the weapons scores produced by APPVUL. Then the APPVUL method requires that those scores be scaled so that R(2) = R(1), where

$$R(2) = \frac{\sum_{j=1}^{N^2} W_j^2 \, V_j^2(2)}{\sum_{i=1}^{N^1} W_i^1 \, V_i^1(2)} \; .$$

However, even with this scaling, $V_i^s(2)$ could be quite different from $V_i^s(1)$ for any or all i ($1 \le i \le N^s$) for both s (s = 1,2).

The APPVUL method is also close to APP in the sense that APPVUL requires that the APP scores, $V_i^s(1)$, be calculated and directly used as part of the calculation of the APPVUL scores $V_i^s(2)$.

Finally, the APPVUL method is close to APP in the following somewhat abstract sense. If one looks at APP as measuring lethality, as opposed to measuring total capability, then APP says that the lethality of a weapon is proportional to the rate at which it could kill enemy weapons times the lethality of those enemy weapons. APPVUL takes this assumption and combines it with a measure of relative invulnerability, where the relative invulnerability of a weapon is assumed to be inversely proportional to the rate at which it could be killed by enemy weapons times the relative invulnerability of those enemy weapons. The overall APPVUL score is then assumed to be the geometric mean of the lethality (i.e., the APP) score and this relative invulnerability score.

In particular, for i = 1,N^s and s = 1,2, let $\hat{V}_i^s$ denote this relative invulnerability score. Then APPVUL assumes that these scores must satisfy the general relationship that

$$\alpha^s \hat{V}_i^s = \begin{cases} \left(\sum_{j=1}^{N^{s'}} \hat{V}_j^{s'} \hat{R}_{ji}^{s'} \right)^{-1} & \sum_{j=1}^{N^{s'}} \hat{V}_j^{s'} \hat{R}_{ji}^{s'} > 0 \\ 0 & \text{otherwise ,} \end{cases}$$

(for some α^s) for $i = 1,N^s$, $s = 1,2$, and $s' = 3-s$, and $\hat{V}^1_h$ must satisfy the scaling relationship that

$$\hat{V}^1_h = 1.$$

However, two scaling assumptions are required (essentially to determine both α^1 and α^2), and the second scaling assumption is constructed as follows. For $i = 1,N^s$ and $s = 1,2$, let $\tilde{V}^s_i$ be a (non-negative) solution to

$$\gamma^s \tilde{V}^s_i = \begin{cases} \left(\sum_{j=1}^{N^{s'}} \tilde{V}^{s'}_i \hat{K}^{s'}_{ji} \right)^{-1} & \sum_{j=1}^{N^{s'}} \tilde{V}^{s'}_j \hat{K}^{s'}_{ji} > 0 \\ 0 & \text{otherwise ,} \end{cases}$$

where

$$\gamma^1 = \begin{cases} \left(\sum_{j=1}^{N^2} \tilde{V}^2_j \hat{K}^2_{jh} \right)^{-1} & \sum_{j=1}^{N^2} \tilde{V}^2_j \hat{K}^2_{jh} > 0 \\ 0 & \text{otherwise} \end{cases}$$

and $\gamma^2 = 1$. Note that this definition of γ^1 gives that

$$\tilde{V}^1_h = 1$$

if $\gamma^1 \neq 0$. (If $\gamma^1 = 0$, then a different value for the input h should be selected.) Setting $\gamma^2 = 1$ is clearly arbitrary, and is done only to obtain preliminary values which can then be scaled so that R(2) = R(1), where R(2) and R(1) are the force ratios defined above. In particular, let

$$\beta = \left[\left(\frac{\sum_{j=1}^{N_2} W_j^2 V_j^2(1)}{\sum_{i=1}^{N^1} W_i^1 V_j^1(1)} \right) \div \left(\frac{\sum_{j=1}^{N^2} W_j^2 \sqrt{\tilde{V}_j^2 V_j^2(1)}}{\sum_{i=1}^{N^1} W_i^1 \sqrt{\tilde{V}_i^1 V_i^1(1)}} \right) \right]^2$$

$$= \left[R(1) / \left(\sum_{j=1}^{N^2} W_j^2 \sqrt{\tilde{V}_j^2 V_j^2(1)} \Big/ \sum_{i=1}^{N^1} W_i^1 \sqrt{\tilde{V}_i^1 V_i^1(1)} \right) \right]^2$$

where V(1) is the APP weapons score and $\tilde{V}$ satisfies the relationships stated just above. Then the final values for the relative invulnerabilities, denoted by $\hat{V}$ above, are determined by setting

$$\hat{V}_i^1 = \tilde{V}_i^1 \qquad i = 1,N^1$$

and

$$\hat{V}_i^2 = \beta \tilde{V}_i^2 \qquad i = 1,N^2 .$$

The overall APPVUL weapons scores are then computed as the geometric mean of the APP score and $\hat{V}$. That is, if V(1) is the APP score and V(2) denotes the APPVUL score, then

$$V_i^s(2) = \sqrt{V_i^s(1)\, \hat{V}_i^s}$$

for $i = 1,N^s$ and $s = 1,2$. Note that $\tilde{V}_h^1 = 1$, and so $\hat{V}_h^1 = 1$. Thus, since $\hat{V}_h^1 = 1$ and $V_h^1(1) = 1$, it follows that $V_h^1(2) = 1$.

The basic idea behind the APPVUL method for calculating weapons scores was proposed in the appendix to Chapter II of [1]. The rationale behind this idea is as follows. Since $\hat{K}_{ji}^{s'}$ gives the rate at which a type-i weapon on side s is being killed by type-j enemy weapons,

$$\sum_{j=1}^{N^{s'}} \hat{V}_j^{s'} \hat{K}_{ji}^{s'}$$

gives a measure of the overall vulnerability of those type-i weapons, where the factor $\hat{V}_j^{s'}$ is applied to the killing capability of enemy weapons of type j to represent that it is worse to be vulnerable to a relatively invulnerable enemy weapon than it is to be vulnerable to a relatively vulnerable one. Accordingly, if the sum above is a measure of relative vulnerability of type-i weapons on side s, then

$$\left(\sum_{j=1}^{N^{s'}} \hat{V}_j^{s'} \hat{K}_{ji}^{s'} \right)^{-1}$$

is a measure of the relative invulnerability of those weapons, which leads to the general relationship concerning invulnerability scores stated above.

3. Potential Exchange Potential

The potential exchange potential (PEXPOT) method for computing weapons scores was also proposed in the appendix to Chapter II of [1], and also uses $\hat{K}$ to add vulnerability considerations to APP. However, it is structurally quite different from APPVUL. In particular, PEXPOT combines lethality and vulnerability into one relationship as follows. PEXPOT essentially assumes that the score of a weapon is proportional to the rate at which that weapon can kill enemy score divided by the rate at which enemy weapons can kill the (friendly) score contributed by that weapon. In particular, if $V_i^s(3)$ denotes the PEXPOT score for weapons of type i on side s, then the PEXPOT approach assumes that V(3) must satisfy the general relationship that

$$\beta V_i^s(3) = \frac{\sum_{j=1}^{N^{s'}} K_{ij}^s V_j^{s'}(3)}{\sum_{j'=1}^{N^{s'}} \hat{K}_{j'i}^{s'} V_i^s(3)}$$

for some β, for i = 1,N^s, s = 1,2, and s' = 3-s, and $V_h^1(3)$ must satisfy the scaling relationship that

$$V_h^1(3) = 1.$$

(Note that the general relationship just above can be rewritten as

$$\beta\left(V_i^s(3)\right)^2 = \sum_{j=1}^{N^{s'}} K_{ij}^s V_j^{s'}(3) \Big/ \sum_{j'=1}^{N^{s'}} \hat{K}_{j'i}^{s'}$$

for i = 1,N^s, s = 1,2, and s' = 3–s.)

4. Lethality/Vulnerability Potential

The lethality/vulnerability potential (LEVPOT) method for computing weapons scores is a new scoring method that attempts to combine selected aspects of the APPVUL and PEXPOT methods. In particular, LEVPOT takes from PEXPOT the idea that weapons scores that consider both lethality and vulnerability can be calculated using only one general relationship by assuming that each such score is proportional to a ratio of terms. (In contrast, APPVUL computes such scores as a mean of the results from two separate relationships.) LEVPOT takes from APPVUL the idea that one of the terms in this ratio should measure the lethality but be independent of the vulnerability of the weapon in question, while the other term should measure the vulnerability but be independent of the lethality of that weapon. Note that PEXPOT does not have this property. In particular, while the numerator of the PEXPOT ratio,

$$n_i^s(3) = \sum_{j=1}^{N^{s'}} K_{ij}^s V_j^{s'}(3) ,$$

does measure the lethality of weapons of type i on side s independently of their vulnerability, the denominator,

$$d_i^s(3) = \sum_{j=1}^{N^{s'}} \hat{K}_{ji}^{s'} V_i^s(3) ,$$

does not measure their vulnerability independently of their lethality. For example, if weapons of types i and i' on side s are such that

$$K_{ij}^s = K_{i'j}^s$$

for all j (so these weapons have equal lethality), then $n_i^s(3) = n_{i'}^s(3)$. However, even if

$$\hat{K}^{s'}_{ji} = \hat{K}^{s'}_{ji'}$$

for all j (so these weapons have equal vulnerability), $d^s_i(3) \neq d^s_{i'}(3)$ in general. Thus, PEXPOT does not give a lethality-to-vulnerability ratio.

LEVPOT does give such a ratio. Since the numerator for PEXPOT does measure lethality independently of vulnerability, the numerator for LEVPOT is the same as that for PEXPOT. Indeed, note that this numerator is the same as the right side of the basic antipotential potential equation given in Section 1, above. The denominator for LEVPOT differs from that of PEXPOT in two ways.

First, the V^s_i factor is removed. It is this factor that causes the denominator of PEXPOT to depend on the lethality of the weapon type being measured. This change alone would produce a lethality-to-vulnerability ratio; however, a second change is also made.

The APPVUL method assumed that it is worse to be vulnerable to a relatively invulnerable enemy weapon than it is to be vulnerable to an enemy weapon that is itself relatively vulnerable. In the APPVUL notation of Section 2, this assumption was formulated as

$$\alpha^s \hat{V}^s_i = 1 \Big/ \sum_{j=1}^{N^{s'}} \hat{V}^{s'}_j \hat{K}^{s'}_{ji} ,$$

assuming that the denominator is greater than zero. That is, $\hat{K}^{s'}_{ji}$ is weighted by $\hat{V}^{s'}_j$. The analogous structure here is to weight $\hat{K}^{s'}_{ji}$ by $V^{s'}_j$, which is the second difference between the denominator in PEXPOT and that in LEVPOT.

Putting these two changes together gives that the denominator used by LEVPOT is:

$$d^s_i(5) = \sum_{j'=1}^{N^{s'}} \hat{K}^{s'}_{j'i} V^{s'}_{j'}(5) ,$$

as opposed to the denominator of PEXPOT:

$$d_i^s(3) = \sum_{j'=1}^{N^{s'}} \hat{K}^{s'}_{j'i} V_i^s(3) .$$

(The reason for indexing LEVPOT by 5 instead of 4 will be explained in Section 5 below.)

As stated above, the numerator in LEVPOT is the same as that in PEXPOT. Thus, if $V_i^s(5)$ denotes the LEVPOT score for weapons of type i on side s, then the LEVPOT approach assumes that V(5) must satisfy the general relationship that

$$\beta V_i^s(5) = \frac{\sum_{j=1}^{N^{s'}} K^s_{ij} V_j^{s'}(5)}{\sum_{j'=1}^{N^{s'}} V^{s'}_{j'}(5) \hat{K}^{s'}_{j'i}}$$

for some β, for $i = 1,N^s$, $s = 1,2$, and $s' = 3-s$, and $V_h^1(5)$ must satisfy the scaling relationship that

$$V_h^1(5) = 1.$$

5. Dynamic Potential

The dynamic potential (DYNPOT) method for computing weapons scores is a new scoring method that was intended to be based on a different approach than APPVUL, PEXPOT, or LEVPOT. However, while the basic assumptions behind DYNPOT are quite different than the assumptions behind the other methods, it turns out that algebraic manipulation reduces DYNPOT to a compromise between PEXPOT and LEVPOT.

The basic assumptions behind DYNPOT are as follows. DYNPOT assumes that the basic flaw in the APP approach is that APP computes an instantaneous score for a weapon, but not a long-run score. DYNPOT further assumes the following: (1) A reasonable score can be computed by assuming that the long-run score of a weapon at the start of combat equals the instantaneous (short-run) score achieved by that weapon plus a measure of the likelihood that the weapon survives the short-run times the long-run score for that weapon from then on in that combat. (2) The APP method gives a reasonable formula for computing the short-run score. (3) A reasonable measure of the likelihood that a weapon of type i on side s survives in the short run is given by

$$1 - \sum_{j=1}^{N^{s'}} \beta \hat{K}^{s'}_{ji}$$

for some $\beta > 0$, where $s' = 3-s$. (4) The long-run score of a weapon does not change over time, so that the long-run score of a weapon measured at the start of combat equals the long-run score for that weapon measured after short-run attrition has been considered. In short, DYNPOT assumes that

(long-run score) = (short-run score) +

(likelihood of surviving the short run) × (long-run score)

In the notation introduced here, if $V^s_i(4)$ denotes the DYNPOT score for weapons of type i on side s, then the assumptions listed above mean that V(4) must satisfy the general relationship that

$$V^s_i(4) = \sum_{j=1}^{N^{s'}} K^s_{ij} V^{s'}_j(4) + \left(1 - \sum_{j=1}^{N^{s'}} \beta \hat{K}^{s'}_{ji}\right) V^s_i(4)$$

for some β, for all $i = 1, N^s$, and for $s = 1,2$. As in the other methods, the scaling assumption that

$$V^1_h(4) = 1$$

is also made.

Note that rearranging terms in the general relationship just above gives that

$$\beta V^s_i(4) = \frac{\sum_{j=1}^{N^{s'}} K^s_{ij} V^{s'}_j(4)}{\sum_{j'=1}^{N^{s'}} \hat{K}^{s'}_{j'i}} \quad .$$

For comparison, the corresponding PEXPOT relationship is

$$\beta V_i^s(3) = \frac{\sum_{j=1}^{N^{s'}} K_{ij}^s V_j^{s'}(3)}{\sum_{j'=1}^{N^{s'}} \hat{K}_{j'i}^{s'} V_i^s(3)} ,$$

while the corresponding LEVPOT relationship is

$$\beta V_i^s(5) = \frac{\sum_{j=1}^{N^{s'}} K_{ij}^s V_j^{s'}(5)}{\sum_{j'=1}^{N^{s'}} V_{j'}^{s'}(5) \hat{K}_{j'i}^{s'}} .$$

Thus, applying the first of the two changes discussed above for converting PEXPOT to LEVPOT converts PEXPOT to DYNPOT, then applying the second change to DYNPOT converts DYNPOT to LEVPOT.

This positioning of DYNPOT between PEXPOT and LEVPOT is the reason for indexing DYNPOT between PEXPOT and LEVPOT. Given that APP is indexed by 1, APPVUL by 2, and PEXPOT by 3, this means that DYNPOT should be indexed by 4 and LEVPOT indexed by 5. This comparison between DYNPOT and LEVPOT is also interesting because it points out a possibly important distinction between them. This distinction is that a weapon might be vulnerable only to enemy weapons which are themselves quite vulnerable. Such a weapon might receive a low DYNPOT score, but not necessarily a low LEVPOT score.

6. Rescaling

Each of the five methods for calculating weapons scores as discussed above has the property that these scores have been scaled so that weapon type h on side 1 receives a score of 1.0, where h is an input to the method. Once calculated, any of these scores could be rescaled in any (appropriate) manner.

One potential problem with the scaling used above is as follows. Suppose two cases are being analyzed using weapons scores. Suppose that h = 1 and that weapons of type 2 on side 1 are expected to receive a much lower score in the second case than in the first, but it is not known what the impact will be on weapons of type 1 on side 1. Suppose it turns out that weapons of type 1 are also worse in the second case than in the first and that the relative decrease is even greater for weapons of type 1 than for those of type 2. Then the scaling structure presented above would reflect this situation by retaining a score of 1.0 for weapons of type 1 on side 1 in both cases (since h = 1), by returning a slightly higher score for weapons of type 2 in the second case than in the first case, and by returning much higher scores for other types of weapons in the second case than in the first. Thus, if one is not careful to look at scores relative to all other weapons, it appears as if the score of weapons of type 2 went up (going from case 1 to case 2) when it should have gone down. In fact, it did go up relative to the scaling weapon (type 1) but not relative to all weapons.

To alleviate this potential problem, COMBAT can rescale all scores so that the score of the average weapon is 1.0. That is, if the scores as calculated above are given by $V_i^s(v)$, where $V_h(v) = 1$, COMBAT can compute the total number of weapons t, where

$$t = \sum_{s=1}^{2} \sum_{i=1}^{N^s} W_i^s \, ,$$

it can compute a scale factor $\alpha(v)$, where

$$\alpha(v) = \sum_{s=1}^{2} \sum_{i=1}^{N^s} V_i^s(v) W_i^s / t \ ,$$

and it can compute rescaled scores $\ddot{V}_i^s(v)$ as

$$\ddot{V}_i^s(v) = V_i^s(v) / \alpha(v)$$

for $i = 1, N^s$, $s = 1,2$. Note that, with this scaling, the overall average weapons score,

$$\sum_{s=1}^{2} \sum_{i=1}^{N^s} \ddot{V}_i^s(v) W_i^s / t \, ,$$

must equal 1.0.

D. AIRCRAFT

Combat aircraft can be addressed in one of two ways in COMBAT. One way is to consider them as composing one or more of the types of weapons systems contained in the N^s types of fully interacting systems being considered. The other way is to consider only ground (CAS) attack aircraft and to consider only the lethality of these aircraft against ground forces. These two approaches are described in the two sections below.

1. The Consideration of Combat Aircraft as Fully Interacting Weapons Systems

In order to consider combat aircraft as composing one or more of the types of weapons systems contained in the N^s types of fully interacting systems being considered for side s, the aircraft must be grouped into types where these types are distinguished both by (possible) differences in air frames and (always) by differences in missions.

For example, suppose that side 1 has 200 F15s, 300 F16s, and 100 A10s. Suppose that 100 F15s are assigned to fly escort missions and 100 are assigned to fly defensive missions. Suppose that 100 F16s are assigned to fly escort missions, 100 to fly defensive missions, and 100 to fly ground attack missions; and suppose that all 100 A10s fly ground attack missions. Then, as one example, side 1 could be considered as having 200 escort aircraft which are a mix of F15s and F16s, 100 defense aircraft which are all F15s, 100 defense aircraft which are all F16s, 100 ground attack aircraft which are all F16s, and 100 ground attack aircraft which are all A10s, for a total of 5 types of aircraft. Splitting the escort aircraft into F15s and F16s would yield 6 types, which is the maximum reasonable number of types in this example. Alternatively, combining all the defenders into one type and all the attackers into another would yield 3 types of aircraft (escort aircraft, defensive aircraft, and ground attack aircraft), which would be the minimum number of types possible in this example since each distinct mission being addressed must have associated with it at least one type of aircraft.

The important point here is that, for aircraft, "type" must distinguish different missions as well as (optionally) distinguishing different air frames. The impact of this structure on using COMBAT as a dynamic model of combat will be discussed further in Chapter VI, below.

2. The Consideration of Ground Attack Aircraft as Non-Vulnerable Weapons Systems

As discussed above, large combat models are frequently composed of various portions and, in some cases, the vulnerabilities of some weapons systems (such as ground attack aircraft) are considered in one portion while major aspects of their lethality (e.g., their lethality against ground weapons) are considered in another portion. In these cases, scores are frequently needed for the non-vulnerable weapons (e.g., attack aircraft) to be used in conjunction with the scores of the fully interacting weapons (e.g., ground weapons) to determine overall output measures of effectiveness, such as force ratios and ground movement (and, perhaps, to help determine tactical decisions and/or other aspects of the combat being modeled).

Of course, one way to obtain such scores is to consider all combat aircraft and all of their missions and just extract the needed scores as appropriate. However, this approach requires significantly more data and could be difficult to implement if the required additional data were not available in a sufficiently consistent form to compute these weapons systems scores.

An alternative approach for developing scores for use in such large combat models is to consider only those aircraft assigned to ground attack missions and to consider these aircraft as being "non-vulnerable" in that their vulnerability is to be considered elsewhere in the combat model. COMBAT can (optionally) consider such non-vulnerable weapons systems as follows.

a. Additional Notation for Non-Vulnerable Weapons Systems

Particular models might consider weapons systems other than aircraft (such as long-range artillery) as being non-vulnerable in the sense described above, or they might consider only a subset of combat aircraft to be non-vulnerable in this sense. The notation below and the COMBAT computer program can easily handle such cases. For ease of discussion only, the term "aircraft" will be used interchangeably with the term "non-vulnerable weapons systems" in the remainder of this chapter. Thus, in the following discussion, "aircraft" is understood to include all types of non-vulnerable weapons systems and to exclude all combat aircraft being considered as part of the N^s types of fully interacting weapons systems on side s.

Assume that the aircraft (i.e., non-vulnerable weapons systems) to be considered on side s have been partitioned into $\underline{N}^s$ types, s = 1,2. Unlike N^s, COMBAT allows $\underline{N}^s$ to equal zero. If $\underline{N}^1 = \underline{N}^2 = 0$, no non-vulnerable systems are to be considered, COMBAT does not expect or accept any of the data denoted below, and all calculations concerning non-vulnerable systems are skipped. If either $\underline{N}^1$ or $\underline{N}^2$ is positive, COMBAT expects the other to be positive also; however, the number of non-vulnerable systems on either side (denoted below by $\underline{W}_i^s$) can be zero. For the remainder of this section and in Section b, below, assume that $\underline{N}^1$ and $\underline{N}^2$ are positive.

Consider the following notation.

$\underline{W}_i^s$ = the number of aircraft of type i on side s, where i = 1,$\underline{N}^s$ and s = 1,2.

$\underline{E}_i^s$ = the average number of engagements per time period (or per unit time) made by an aircraft of type i on side s (against all vulnerable enemy weapons), where $i = 1,\underline{N}^s$ and $s = 1,2$.

$\underline{P}_{ij}^s$ = the probability of kill per engagement by an aircraft of type i when that aircraft is engaging an enemy weapon of type j, where $i = 1,\underline{N}^s$, $j = 1,N^{s'}$, $s = 1,2$, and $s' = 3-s$.

$\tilde{\underline{A}}$ = a set of input parameters used in conjunction $\underline{E}$, $\underline{P}$, W, E, and P to calculate allocations of fire. (Specifications for $\tilde{\underline{A}}$ will be discussed in Chapter III.)

$\underline{F}_{ij}^{as}$: For each a (the range of a is discussed in Chapter III), $\underline{F}_{ij}^{as}$ is a function that maps $\underline{E}$, $\underline{P}$, W, E, P, and $\tilde{\underline{A}}$ into the average fraction of engagements (i.e., the allocation of fire) that aircraft of type i on side s make against enemy weapons of type j (out of all of the engagements made by those type-i aircraft), where $i = 1,\underline{N}^s$, $j = 1,N^{s'}$, $s = 1,2$, and $s' = 3--s$. (Various formulas for $\underline{F}^a$ will be discussed in Chapter III.)

$\underline{A}_{ij}^s$ = $\underline{A}_{ij}^s(a) = \underline{F}_{ij}^{as}(\underline{E}, \underline{P}, W, E, P, \tilde{\underline{A}})$, where $i = 1,\underline{N}^s$, $j = 1,N^{s'}$, $s = 1,2$, and $s' = 3-s$.

$\underline{K}_{ij}^s$ = $\underline{E}_i^s \, \underline{A}_{ij}^s \, \underline{P}_{ij}^s$, where $i = 1,\underline{N}^s$, $j = 1,N^{s'}$, $s = 1,2$, and $s' = 3-s$.

$\underline{V}_i^s(v)$ = the lethality score, as calculated using scoring method v, for aircraft of type i on side s, where $i = 1,\underline{N}^s$ and $s = 1,2$.

b. The Calculation of Lethality Scores for Non-Vulnerable Weapons Systems

Note that, except for PEXPOT, each of the scoring methods described in Section C above use the term

$$\sum_{j=1}^{N^{s'}} K_{ij}^s \, V_j^{s'}(v)$$

to directly measure the lethality of weapons of type i on side s, and that PEXPOT also uses this term in the numerator of its general scoring relationship (as a partial measure of this lethality). Thus, a consistent approach to measure the lethality of aircraft is to assume that $\underline{V}_i^s(v)$ is proportional to the term

$$\sum_{j=1}^{\underline{N}^{s'}} \underline{K}_{ij}^s \, V_j^{s'}(v) \, ,$$

where $V_j^{s'}(v)$ is the score for weapons of type j on side s' as calculated using scoring method v by considering only fully interacting weapons in that scoring method, where i = $1,\underline{N}^s$, s = 1,2, and s' = 3–s.

Except for APPVUL, each of the scoring methods described in Section C above computes a proportionality factor, denoted by β in each of the corresponding subsections of Section C. Since these βs are, in general, not equal to each other, let v index β here. Thus β(v) corresponds to the value of β that would be computed by scoring method v for v = 1, 3, 4, and 5. APPVUL computes several proportionality factors, but one, which must equal β(1), is used to scale lethality. Thus, setting β(2) = β(1) gives a reasonable value of β for APPVUL.

Using these scaling factors, $\underline{V}(v)$ can be directly computed by the formula

$$\underline{V}_i^s(v) = \left(\sum_{j=1}^{\underline{N}^{s'}} \underline{K}_{ij}^s \, V_j^{s'}(v) \right) / \, \beta(v)$$

for all scoring methods v, for i = $1,\underline{N}^s$, s = 1,2, and s' = 3–s.

Rescaled aircraft scores can be calculated using the same rescaling factors, α(v), defined in Section C.6 above. Using these factors, rescaled aircraft scores for scoring method v, $\dot{\underline{V}}(v)$, are calculated by the formula

$$\dot{\underline{V}}_i^s(v) = \underline{V}_i^s(v) \, / \, \alpha(v)$$

where i = $1,\underline{N}^s$ and s = 1,2. Note that, with this rescaling, the overall average weapons score not including aircraft remains at 1.0. However, since the reason for considering rescaling described in Section C.6 cannot occur here, this structure is completely consistent with the rationale underlying that section.

E. ALTERNATIVE MEASURES FOR COMPARING FORCES

1. Background

As indicated above, static comparisons of two opposing forces are frequently formed in the following fashion. First, the resources in each force are grouped into types, then scores are assigned to resources by these types, and force strengths are calculated as

$$S^s = \sum_{i=1}^{N^s} V_i^s W_i^s$$

where N^s is the number of types of resources in the force belonging to side s, W_i^s is the number of resources of type i in that force, V_i^s is the score assigned to each resource of type i in that force, and S^s is the resulting force strength of side s. (Of course, other methods exist for calculating S^s as a function of $V_1^s,\ldots,V_N^s, W_1^s,\ldots,W_N^s$. One such set of methods, with particular applicability to PEXPOT, LEVPOT, and DYNPOT, is discussed in Appendix E.) Given S^1 and S^2, a force ratio R can be calculated as $R = S^2/S^1$, and the force ratio R^{-1} can be calculated as $R^{-1} = S^1/S^2$. This structure applies to simple "bean counts" as well as to more complex approaches that utilize the methods suggested in Section C above. For example, one way to picture a "bean count" of tanks is as follows. $N^1 = N^2 = 2$, a resource is of type 1 if it is to be called a tank, and it is of type 2 otherwise, so that W_1^s gives the number of resources that are to be called tanks on side s while W_2^s gives the number of all of the other resources on side s. Then, setting $V_1^s = 1.0$ and $V_2^s = 0.0$ gives force strengths and force ratios according to a "bean-count" of tanks.

Dynamic comparison of two opposing forces are frequently constructed in an analogous fashion. For example, suppose that T time periods are to be considered, let W_{it}^s denote the number of resources at the (start or) end of time period t, and let V_{it}^s denote the score of resources of type i during time period t. Assuming that attrition is occurring, W_{it}^s will not be constant with respect to t. Whether or not attrition is occurring, V_{it}^s could be assumed to be constant (with respect to t) by being calculated using average or typical force effectiveness parameters, or V_{it}^s could depend on t by being recalculated each time period. In either case, the resulting force strength,

$$S_t^s = \sum_{i=1}^{N^s} V_{it}^s W_{it}^s ,$$

will depend on t since W_{it}^s depends on t. An overall dynamic comparison of two forces can then be calculated as $f(g_1(S_1^1,S_1^2),...,g_T(S_T^1,S_T^2))$ for suitably defined functions f and g_1 through g_T. For simplicity, assume that $f(x_1,...,x_T) = x_1 + ... + x_T$ and that $g_1(x,y) = ... = g_T(x,y) = g(x,y)$--these are commonly made assumptions, and COMBAT makes these assumptions. Then this overall dynamic comparison reduces to

$$\sum_{t=1}^{T} g(S_t^1,S_t^2) .$$

In this form, the static force ratio measure discussed above is the special case of this dynamic measure in which T = 1 and g(x,y) is either x/y or y/x. Note, however, that setting g(x,y) equal to x/y or to y/x does not make sense if $T \geq 2$. For example, if side 1 were twice as strong as side 2 on day 1, and if side 2 were twice as strong as side 1 on day 2, then a reasonable net comparison of these forces might be somewhere near equality, not near $2\frac{1}{2}$. Dynamic models of combat frequently address this modeling issue by setting $g(S_t^1,S_t^2)$ to be an estimate of the average distance that the opposing forces would move during time period t when the strengths of these forces are given by S_t^1 and S_t^2, respectively.

COMBAT does not attempt to directly estimate the movement of forces. Instead, COMBAT provides several other alternative specifications for the function denoted by g above.

2. Structure

First it should be noted that, for each time period being considered, COMBAT computes and displays both

$$S_t^2(v) / S_t^1(v) \text{ and } S_t^1(v) / S_t^2(v) ,$$

where $S_t^s(v)$ is the force strength calculated using scoring method v, i.e.,

$$S_t^s(v) = \sum_{i=1}^{N^s} V_{it}^s(v) W_{it}^s .$$

However, essentially for the reasons discussed above, COMBAT does not compute

$$\sum_{\tau=1}^{t} S_\tau^2(v) / S_\tau^1(v)$$

for any t (greater than one), nor does it compute the sum of the inverses of these ratios. Instead, COMBAT computes and displays both

$$g_{(m)}(S_t^1(v), S_t^2(v))$$

and

$$\sum_{\tau=1}^{t} g_{(m)}(S_\tau^1(v), S_\tau^2(v))$$

for all t from 1 through T for certain functions $g_{(m)}$. These $g_{(m)}$ are all of the form

$$g_{(m)}(x,y) = \begin{cases} \dfrac{y-x}{m(x,y)} & m(x,y) > 0 \\ +\infty & m(x,y) = 0,\ x = 0,\ y > 0 \\ -\infty & m(x,y) = 0,\ x > 0,\ y = 0 \\ 0 & \text{otherwise}, \end{cases}$$

where m(x,y) is a mean of x and y. (The COMBAT computer code uses 10^9 for $+\infty$ and -10^9 for $-\infty$.)

3. Means

Let the interval $[0,\infty)$ be denoted by $\mathbb{R}^+$, and let m denote a function that maps $\mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$. Then m is called a (two-variable) mean if and only if m satisfies the following four properties for all x and y in $\mathbb{R}^+$.

(1) Mean Property: $\min(x,y) \leq m(x,y) \leq \max(x,y)$.

(2) Symmetry: $m(x,y) = m(y,x)$.

(3) Homogeneity: $m(cx,cy) = cm(x,y)$ for all c in $\mathbb{R}^+$.

(4) Monotonicity: If $u \geq x$ and $v \geq y$ then $m(u,v) \geq m(x,y)$.

Some typical examples of such means are as follows. The arithmetic mean is given by

$$m_a(x,y) = (x + y) / 2 .$$

The geometric mean is given by

$$\dot{m}_g(x,y) = \sqrt{xy} \ .$$

The harmonic mean is given by

$$m_h(x,y) = \begin{cases} 1 / ((1/x + 1/y) / 2) & x > 0,\ y > 0 \\ 0 & \text{otherwise}, \end{cases}$$

$$= \begin{cases} 2xy/(x + y) & x > 0,\ y > 0 \\ 0 & \text{otherwise} . \end{cases}$$

The logarithmic mean is given by

$$m_l(x,y) = \begin{cases} (y - x) / \ln(y/x) & x > 0,\ y > 0,\ x \neq y \\ x & x = y \\ 0 & \text{otherwise} . \end{cases}$$

Further, note that

$$m_n(x,y) = \min(x,y)$$

and

$$m_x(x,y) = \max(x,y)$$

are also means according to this definition.

Means can be classified in several different ways. Of some interest here is whether these means satisfy the

$$\text{Null Property: } \lim_{x \downarrow 0} m(x,y) = 0 \qquad \text{for all } y \text{ in } \mathbb{R}^+ ,$$

and whether they satisfy the

$$\text{Bounded Property: } \lim_{x \to \infty} m(x,y) < \infty \qquad \text{for all } y \text{ in } \mathbb{R}^+.$$

For the six means listed above, the following taxonomy holds.

	Bounded	**Non-bounded**
Null	minimum (m_n) harmonic (m_h)	geometric (m_g) logarithmic (m_l)
Non-null		arithmetic (m_a) maximum (m_x)

Note also that

$$m_n(x,y) \leq m_h(x,y) \leq m_g(x,y) \leq m_l(x,y) \leq m_a(x,y) \leq m_x(x,y)$$

for all x and y in $\mathbb{R}^+$, and that all of these inequalities are strict if $x > 0$, $y > 0$, and $x \neq y$.

4. Measures

If y is constant and strictly positive then, as $x \downarrow 0$,

$$g_a(x,y) = \frac{y-x}{m_a(x,y)} \to 2$$

and

$$g_x(x,y) = \frac{y-x}{m_x(x,y)} \to 1 \; ;$$

whereas

$$g_n(x,y) = \frac{y-x}{m_n(x,y)} \; ,$$

$$g_h(x,y) = \frac{y-x}{m_h(x,y)} \; ,$$

$$g_g(x,y) = \frac{y-x}{m_g(x,y)} \; , \text{ and}$$

$$g_l(x,y) = \frac{y-x}{m_l(x,y)}$$

all diverge to infinity. Note also that the same limits apply as $y \to \infty$ if x is constant and strictly positive. Thus, only the latter four of these six measures have the property that, as one of the forces being considered becomes much smaller than the other (and the other remains relatively stable), then the comparative measure of these forces grows without bound. Further, only the latter four of these six measures have the property that, as one of the forces being considered grows much larger than the other (and the other remains relatively stable), then the comparative measure of these forces also grows without bound. There may be special cases for which one does not want either or both of these properties to hold. In addition, there certainly are many means (i.e., functions from $\mathbb{R}^+ \times \mathbb{R}^+$ to $\mathbb{R}^+$ satisfying the mean, symmetry, homogeneity, and monotonicity properties listed above) other than the examples given here (see, for example, [2]), and measures based on these other means may or may not have one or both of these properties. However, for the time being, COMBAT considers only the latter four measures listed above. Accordingly, COMBAT computes and displays:

$$\begin{aligned}
&g_n(S_t^1(v), S_t^2(v)), \qquad && \sum_{\tau=1}^{t} g_n(S_\tau^1(v), S_\tau^2(v)), \\
&g_h(S_t^1(v), S_t^2(v)), && \sum_{\tau=1}^{t} g_h(S_\tau^1(v), S_\tau^2(v)), \\
&g_g(S_t^1(v), S_t^2(v)), && \sum_{\tau=1}^{t} g_g(S_\tau^1(v), S_\tau^2(v)), \\
&g_l(S_t^1(v), S_t^2(v)), \text{ and } && \sum_{\tau=1}^{t} g_l(S_\tau^1(v), S_\tau^2(v)),
\end{aligned}$$

for all t from 1 through T, where g_n, g_h, g_g, and g_l are as defined above.

5. References

The structure used here is based on the research reported in References [3] and [4]. Reference [3] considers the general problem of measuring the relative difference of two quantities, it introduces the general form

$$g_{(m)}(x,y) = \frac{y-x}{m(x,y)},$$

for measuring this relative difference, it lists a host of such measures (including all those discussed above), and it recommends using the logarithmic measure denoted by

$$g_l(x,y) = \frac{y-x}{m_l(x,y)}$$

above.

As an aside, it should be noted that [3] does not use the same definition for a two-variable mean as that given here. In particular, [3] keeps the mean, symmetry, and homogeneity properties, but it replaces the monotonicity property with the

Continuity Property: m is a continuous function.

It is easy to construct functions that map $\mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ that satisfy the mean, symmetry, and homogeneity properties and that:

(a) satisfy neither monotonicity nor continuity, or

(b) satisfy monotonicity but not continuity, or

(c) satisfy continuity but not monotonicity.

Thus, the definition of a mean given in [3] is not equivalent to the definition given here. The definition given in [3] is the typical one (see, for example, [2] and [5]), and the choice between these definitions may not be very important for two-variable means--all of the means discussed above (and all others discussed in [3]) satisfy both monotonicity and continuity. However, when defining means of three or more variables: (a) it can be important to assume that at least one of these two properties (monotonicity or continuity) holds, (b) one can invent cases in which the use of a monotonic but discontinuous mean seems appropriate, (c) for those cases in which continuity is also desired, continuous means can be defined as being those means (as defined here) that also satisfy the continuity property, but (d) it is not at all clear what reasonable use could be made of continuous but non-monotonic "means." It is this rationale that motivates the definition of a mean given here.

Chapters I and II of Reference [4] study a significant subset of noncooperative, finite, two-player (bimatrix) games; namely, those games in which, if one player does better, the other player must necessarily do worse but not necessarily by the same amount (where better and worse are determined by comparing the payoffs that result from playing two alternative pairs of pure strategies). Chapter III of [4] argues that a subset of these games often can be reasonably converted into zero-sum games, essentially by constructing payoffs using g_n as defined above.

The important point here is as follows. It is frequently appropriate to consider a simulation of combat as representing a noncooperative two-player game between the two sides involved. Many of the results of such situations either (a) are arrays (such as the numbers of all of the various types of weapons systems), not single-valued (real) numbers, or (b) are not comprehensive (such as the number of a particular type of weapons system), or (c) do not yield zero-sum payoffs (such as using simple force ratios calculated, for example, by

$$\sum_{\tau=1}^{T} S_{\tau}^{2}(v) \,/\, S_{\tau}^{1}(v)$$

for any scoring method v.) However, for any mean m, if x represents a comprehensive measure of the strength of side 1's force and y such a measure of side 2's force, then $g_{(m)}(x,y)$ will yield comprehensive zero-sum payoffs by setting the payoff to side 2 equal to $g_{(m)}(x,y)$ and the payoff to side 1 equal to $-g_{(m)}(x,y)$.

In particular, for scoring method v, COMBAT computes and displays:

$$\begin{aligned}
&g_{n}(S_{T}^{1}(v),\, S_{T}^{2}(v))\,, && \sum_{\tau=1}^{T} g_{n}(S_{\tau}^{1}(v),\, S_{\tau}^{2}(v))\,,\\
&g_{h}(S_{T}^{1}(v),\, S_{T}^{2}(v))\,, && \sum_{\tau=1}^{T} g_{h}(S_{\tau}^{1}(v),\, S_{\tau}^{2}(v))\,,\\
&g_{g}(S_{T}^{1}(v),\, S_{T}^{2}(v))\,, && \sum_{\tau=1}^{T} g_{g}(S_{\tau}^{1}(v),\, S_{\tau}^{2}(v))\,,\\
&g_{\ell}(S_{T}^{1}(v),\, S_{T}^{2}(v))\,, \text{ and} && \sum_{\tau=1}^{T} g_{\ell}(S_{\tau}^{1}(v),\, S_{\tau}^{2}(v))\,,
\end{aligned}$$

all of which are comprehensive measures and all of which yield zero-sum payoffs. Thus, using any one of these measures, COMBAT can be considered as evaluating the payoff (for any particular pair of pure strategies) of a two-player zero-sum game between side 1 and side 2. Accordingly, this structure allows direct application of the relatively powerful results of two-player zero-sum game theory to analyses that use COMBAT (or that use other simulations whose payoffs are constructed using measures of force effectiveness like those reported here).

F. SETTING WEAPONS SCORES BY DIRECT INPUT

In addition to the options described above for computing weapon scores, COMBAT also allows the option of setting (fully interacting) weapons scores directly equal to input values. As a minor point, this option allows the force comparison measures described above to be determined by input weapon scores. However, the major reason for allowing scores to be determined by direct input is so that input scores can be used in conjunction with allocations of fire (computed using one of the methods described in Chapter III or in Chapter IV, below) and with attrition assessments (computed using one of the methods described in Chapter V, below) to yield a dynamic model of combat whose output measures are determined, in part, by these input scores. Note that COMBAT is truly a dynamic model of combat only if all weapons systems being considered are (at least potentially) vulnerable to enemy fire. Accordingly, if COMBAT is being used as a static model in that the vulnerability of some weapons systems (e.g., aircraft) is not simulated in the model, then COMBAT must attempt to calculate weapons scores. That is, if $\underline{N}^s > 0$ for either s (as discussed in Section D.2.a, above), then COMBAT cannot set weapons scores equal to input values.

A reasonable way to use COMBAT as a dynamic model, or to use any generally similar dynamic model of combat, is as follows. When analyzing a set of issues using such a model, first construct a "typical-case" data set. These typical-case data should contain a sufficiently full span of weapons systems and associated effectiveness parameters so that all of the types of weapons systems to be considered on each side in the analysis are appropriately represented. (In particular, no weapons system to be considered in the analysis should be at zero strength, and none should have absolutely no effectiveness or have an uncharacteristically high effectiveness in this typical-case data set.) Next, a one-time calculation should be made using an allocation method such as one of those described in Chapters III and IV below, and using a scoring method (not direct input) such as one of those described above, to make a one-time computation of weapons scores for this typical case. (Since the goal of this step is only to compute typical weapons scores, attrition need not be calculated or assessed in this one-time computation.) The issues to be analyzed would then be addressed using COMBAT, or using another dynamic model of combat, where (with one class of exceptions) weapons scores are always set by input directly equal to these typical-case scores. The one class of exceptions concerns sensitivity analyses in which different sets of typical scores could be calculated and used to see if these different sets of input scores substantially change the conclusions of the analysis.

The important characteristics of this approach are as follows. First, it does not require that weapons scores be obtained from exogenous and perhaps inconsistent sources--weapons scores are determined internally, are consistent with each other, are consistent with other data being used in the analysis, and are comprehensive in that relative capabilities of all relevant weapons systems are addressed. Second, these scores, once computed, remain fixed both over all time periods in any one run of the model and (except for explicitly constructed sensitivity analyses) over all runs of the model made in the analysis. In a sense, this approach uses weapons scores in a static manner (in that they remain fixed throughout) to do what static scores are reasonably good at doing--helping to summarize the relative effectiveness of two opposing forces at a particular point in time (i.e., either initially or, when running a dynamic combat model, after any given time period). Moreover, this approach uses the dynamic portion of the combat model (not weapons scores) to assess the impact of dynamic interactions. COMBAT has been explicitly designed to allow (as an option) this type of approach for using a dynamic combat model.

III. ALLOCATIONS OF FIRE THAT DO NOT DEPEND ON WEAPONS SCORES

A. INTRODUCTION

1. Background

This chapter discusses specifications for sets $\tilde{A}^a$ and for functions F^a to map W, E, P, and $\tilde{A}^a$ into the average fraction of engagements (i.e., allocation of fire) that weapons of type i on side s make against enemy weapons of type j (out of all of the engagements made by that type of weapon), where $i = 1,N^s$, $j = 1,N^{s'}$, $s = 1,2$, and $s' = 3-s$, and where all of this notation is as defined in Section B of Chapter II, above.

Note that "allocation of fire" here refers to an average fraction of engagements made by each type of (shooting) weapon on one side against each type of (target) weapon on the other side out of all engagements made by that type of (shooting) weapon. Thus, setting

$$A_{ij}^s(a) = F_{ij}^{as}(W,E,P,\tilde{A}^a) = F_{ij}^{as}(W,E,P,\tilde{A}) \, ,$$

it follows that

$$0 \le A_{ij}^s(a) \le 1$$

for each allocation method (indexed by a) being considered, each i from 1 to N^s, each j from 1 to $N^{s'}$ and both s (throughout, $s' = 3-s$). Further,

$$\sum_{j=1}^{N^{s'}} A_{ij}^s(a) = \begin{cases} 0 & \text{if, against the enemy force in question,} \\ & \text{weapons of type i engage no targets} \\ 1 & \text{otherwise} \end{cases}$$

for all such a, i, and both s. Accordingly, if $N^{s'} = 1$ and if weapons of type i will attempt to engage enemy targets if any are present, then the specification of F_{ij}^{as} is trivial in that this specification is needed only for $j = 1$ and

$$F^{as}_{ij}(W,E,P,\tilde{A}) = \begin{cases} 1 & W^{s'}_{1} > 0 \\ 0 & \text{otherwise} \end{cases}$$

for all a, all i, and all W, E, P, and $\tilde{A}$ (and so, in this case, there is no need to specify $\tilde{A}$).

If $N^{s'} > 1$, a simple and mathematically consistent specification of F is to set F_{ij} equal to a constant, say c_{ij}, independent of W, E, P, and $\tilde{A}$, provided only that $W^{s'}_{j'} > 0$ for all j'; i.e.,

$$F^{a}_{ij}(W,E,P,\tilde{A}) = \begin{cases} c_{ij} & W^{s'}_{j'} > 0,\ j' = 1,N^{s'} \\ 0 & W^{s'}_{j} = 0 \\ \tilde{c}_{ij} & \text{otherwise}\ , \end{cases}$$

where the c_{ij} are nonnegative and satisfy

$$\sum_{j=1}^{N^{s'}} c_{ij} = 1\ ,$$

and where $\tilde{c}_{ij}$ is some function of c and $W^{s'}$ that satisfies the conditions that $\tilde{c}_{ij}$ is nonnegative and

$$\sum_{\substack{j=1 \\ W^{s'}_{j} > 0}}^{N^{s'}} \tilde{c}_{ij} = 1\ .$$

In addition to being relatively simple to state, this specification has mathematical characteristics that can significantly simplify calculations, models, and analyses that are based on this specification. Perhaps because of these reasons, this specification has occasionally been used, especially in models that extend the homogeneous Lanchester square attrition equation

$$\Delta W^{s'}_{1} = -\min\{c_{11}k_{11}W^{s}_{1}, W^{s'}_{1}\} \qquad \text{for } s = 1,2$$

to the heterogeneous version given by

$$\Delta W^{s'}_{j} = -\min\{\sum_{i=1}^{N^{s}} c_{ij}k_{ij}W^{s}_{i}, W^{s'}_{j}\} \qquad \text{for } j = 1,N^{s'} \text{ and } s = 1,2\ ,$$

where k_{ij} denotes a probability-of-kill for weapons of type i against enemy weapons of type j.

Unfortunately, besides being simple, computationally attractive, and occasionally used, this specification is completely inappropriate as a method to simulate the allocation of fire in conventional combat; i.e., it is too simplistic to be meaningfully used. The problem here lies in assuming that, except for vacuous cases, the allocation of fire to targets by type is independent of the numbers of targets of each type present. For example, it is ludicrous to believe that weapons of any type would, on average, engage enemy tanks at the same rate if the enemy force had 200 tanks and 2 other armored vehicles as they would if the enemy force had 2 tanks and 200 other armored vehicles. With one exception (discussed next), it is so meaningless to allocate fire by fixed percentages (as described above) that it is not reasonable to investigate the impact of such allocations even for theoretical or methodological purposes and (with this one exception) COMBAT does not allow (even as an option) allocating fire in this way.

The one exception, noted just above considers allocating fire according to a strict priority among detected and/or engageable targets. The important points here are twofold. First, it is just as unreasonable to use strict priority allocation as it is to use other fixed percentage allocations in studies of major defense issues; however, unlike other fixed percentage allocations, it can be reasonable to investigate the impact of strict priorities for theoretical purposes. Second, it is quite reasonable (indeed, it is important) to consider priorities along with other factors (such as the numbers of weapons by type present) when determining allocations of fire. The problem here concerns using priorities in a strict manner, as described next.

Generally, rules to allocate fire by a strict priority proceed as follows. Initially, each type of shooting weapon is given a priority of fire over the various types of target weapons. Then, for each shooting weapon in each "battle," some subset of the targets is selected as being the set of targets that are detected and engageable by that shooting weapon in that battle. Next, that shooting weapon fires at one of the targets that is of the highest priority type (according to the initial priority list for that type of shooter) among the types of targets that the shooter has detected and can engage. Finally, given this shooter/target combination, attrition is assessed. See [6] for a simple example of how such a priority rule can be incorporated into a particular attrition process.

There is nothing intrinsically wrong with investigating theoretical aspects of such allocations of fire, and equations based on such allocations could be developed and incorporated into models such as COMBAT as options for tests and special cases.

However, there are several problems with requiring that all allocations of fire be made by strict priority.

All other things being equal, it may be quite reasonable to allocate fire by strict priorities. But all other things may never be equal. In detailed models, much more is usually known about the situation than just the type of shooter and types of potential targets involved--and these other factors (discussed below) should also be taken into account. In aggregated (e.g., theater level) models, perhaps only the types of shooters and types of potential targets are known to the model. Nevertheless, in reality, the other factors are still there and should, at least implicitly, be taken into account. For example, an otherwise low priority target may become high priority if (a) it is immediately threatening the shooter or other weapons on the shooter's side, or (b) it is occupying particularly valuable territory that the shooter desires to capture (or neutralize). Conversely, an otherwise high priority target may drop in priority if it is not currently threatening friendly weapons or if firing at it would, due to the particular situation, put the shooter in greater jeopardy. These considerations imply (and examination of results seem to bear out) that allocation of fire in real combat and in detailed models do not adhere to strict priorities. Accordingly, adhering to strict priorities is likewise inappropriate in aggregated models.

Another problem with always adhering to strict priorities is as follows. The procedure for determining whether a particular enemy weapon is detected and engaged by a particular shooter might be (and frequently is--see, for example, [7]) independent of the numbers of other enemy weapons on the battlefield. Thus, a model that uses a strict priority procedure might assume that an antitank wepaon is equally able to detect and engage an enemy tank if there are 20 enemy tanks and 20 enemy armored personnel carriers (APCs) on the battlefield as it would be if there were 20 enemy tanks and 2,000 enemy APCs in the battle. Yet it is quite reasonable to believe that, even if an antitank weapon has tanks as its first priority, such a weapon will fire fewer rounds against tanks when the enemy force consists of 20 tanks and 2,000 APCs than it will fire against tanks if the enemy force consists of 20 tanks and 20 APCs. For instance, in a force with 20 tanks and 2,000 APCs, some APCs might obscure some tanks, so that tanks that would have been detectable and engageable in a 20-tank and 20-APC force would not be so in a 20-tank and 2,000-APC force. Also, some APCs might be in the general line of fire between some shooters and some detected tanks, so that those shooters must fire on APCs in between. Strict priority rules (see [6] and [7]) assume that, since there are 20 tanks in both cases, tanks would be fired on equally often in both cases.

The allocation procedures discussed below and in Chapter IV can take priorities into account, but they do so in ways that do not require strict priority rules to be followed in all cases. In particular, except for optional special cases, these procedures allow each enemy weapon to have some (perhaps small) probability of being fired upon, even if engageable higher valued targets are also present. Accordingly, the allocation procedures discussed here are generally more realistic than allocation procedures that require shooters to strictly follow input priorities. (Of course, in models more detailed than COMBAT, there may be particular cases, such as when an aircraft on an air base attack mission detects both a shelter and a nonsheltered aircraft on the ground, where strict priorities might reasonably apply. In such cases, care should be taken to use rules appropriate for the interaction in question.)

The arguments presented above can be stated in mathematical form as follows. For each allocation method a, F^a must satisfy the following properties. (For simplicity in the notation below, $\tilde{A}$ is written in place of $\bar{\tilde{A}}^{a}$.)

First, it must be a valid fractional allocation. That is, as indicated above,

$$0 \le F_{ij}^{as}(W,E,P,\tilde{A}) \le 1$$

and

$$\sum_{j=1}^{N^s} F_{ij}^{as}(W,E,P,\tilde{A}) = \begin{cases} 0 & \text{if, according to } (W,E,P,\tilde{A}), \text{ weapons of type } i \text{ engage no targets} \\ 1 & \text{otherwise} \end{cases}$$

for all relevant i, j, and s.

Second, it must be able to consider priorities. That is, F^a must be structured in such a way that, for each relevant i and j, and each $\tilde{A}$ such that

$$0 < F_{ij}^{as}(W,E,P,\tilde{A}) < 1 \ ,$$

there must exist an $\tilde{A}'$ and an $\tilde{A}''$ such that

$$F_{ij}^{as}(W,E,P,\tilde{A}') > F_{ij}^{as}(W,E,P,\tilde{A}) > F_{ij}^{as}(W,E,P,\tilde{A}'')$$

yet (if $N^s > 1$) an i' exists such that

$$F_{i'j}^{as}(W,E,P,\tilde{A}') = F_{i'j}^{as}(W,E,P,\tilde{A}) = F_{i'j}^{as}(W,E,P,\tilde{A}'')$$

In this case, $\tilde{A}'$ gives a higher priority (and $\tilde{A}''$ a lower priority) for weapons of type i to fire at targets of type j than does $\tilde{A}$, without changing priorities for weapons of type i'.

Third, except for special cases, it must not be a fixed percentage or strict priority allocation. That is, F^a must have the following property. Given $W^{s'}$, where

$$W^{s'} = (W_1^{s'},\ldots,W_{N^{s'}}^{s'}) ,$$

let $W^{s'}(j)$ be defined by

$$W^{s'}(j) = (W_1^{s'},\ldots,W_{j-1}^{s'},W_j^{s'}+1,W_{j+1}^{s'},\ldots,W_{N^{s'}}^{s'}) .$$

That is, the force denoted by $W^{s'}(j)$ is the same as the force denoted by $W^{s'}$, except that it has one more weapon of type j. Then, given any force $W^{s'}$, there must exist an $\tilde{A}$ such that F^a satisfies the inequalities

$$F_{ij}^{as}(W(j),E,P,\tilde{A}) > F_{ij}^{as}(W,E,P,\tilde{A}) > F_{ij}^{as}(W(j'),E,P,\tilde{A})$$

and

$$F_{ij'}^{as}(W(j'),E,P,\tilde{A}) > F_{ij'}^{as}(W,E,P,\tilde{A}) > F_{ij'}^{as}(W(j),E,P,\tilde{A})$$

for all relevant i, j, j'($j \neq j'$), and both s. An $\tilde{A}$ for which any of these inequalities is not strictly satisfied is a "special case" in the sense defined above, and the condition here is that not all $\tilde{A}$ be "special cases." Note that, to avoid logically perverse cases, it must be that

$$F_{ij}^{as}(W(j),E,P,\tilde{A}) \geq F_{ij}^{as}(W,E,P,\tilde{A}) \geq F_{ij}^{as}(W(j'),E,P,\tilde{A}) ,$$

and

$$F_{ij'}^{as}(W(j'),E,P,\tilde{A}) \geq F_{ij'}^{as}(W,E,P,\tilde{A}) \geq F_{ij'}^{as}(W(j),E,P,\tilde{A}) ,$$

and so the meaningful issue is whether or not equalities occur here.

All of the allocation methods described below in this chapter and in Chapter IV satisfy all three of these properties.

2. Numbering These Allocation Methods

This chapter describes five different methods for determining allocations of fire that are independent of weapons scores. Following this, Chapter IV describes four methods for determining allocations that are dependent on weapons scores. Four of the five methods described below are directly analogous to the four methods described in Chapter IV, and these methods are referred to as Methods 1 through Methods 4 in each chapter (i.e., for a = 1 through a = 4, allocation method a below is directly analogous to allocation method a in Chapter IV.)

The allocation method discussed in Section B, below, has no direct counterpart in Chapter IV. This method is described first because it is relatively simple and because it is needed as part of the calculations used by Methods 1, 2, and 3 of both chapters. Since it is described first, it is referred to as Method 0.

Methods 0, 1, 2, and 3 described below, and Methods 1, 2, and 3 of Chapter IV, have been coded and are available for use in the COMBAT computer program. Method 4 of this chapter and Method 4 of Chapter IV have not been coded and are not available for use--their descriptions are given here only to provide examples of additional methods to allocate fire that satisfy the properties stated in Section 1, above.

B. METHOD 0: ALLOCATIONS THAT ARE PROPORTIONAL TO WEIGHTED NUMBERS OF TARGETS

A relatively simple scheme for determining allocations of fire is obtained by assuming that the probability that any particular weapon fires at any particular type of enemy weapon (given that it will fire at some enemy weapon) is proportional to the weighted number of weapons of that type present in the battle, where these weights are determined via inputs. In particular, for i = 1,N^s and s = 1,2, let C^s_{ij} be a set of j weighting factors (for j = 1,$N^{s'}$) used to determine the allocation of fire of weapons of type i on side s. Then Method 0 assumes that the probability that any particular weapon of type i on side s engages a particular enemy weapon of type j, given that it will engage some enemy weapons, is:

$$C^s_{ij} / \sum_{j'=1}^{N^{s'}} C^s_{ij'} W^{s'}_{j'} .$$

Thus, the average fraction of engagements that weapons of type i on side s make against enemy weapons of type j (out of all of the engagements used by that type of weapon) according to Method 0, $A^s_{ij}(0)$, is given by

$$A^s_{ij}(0) = \frac{C^s_{ij} W^{s'}_j}{\sum_{j'=1}^{N^{s'}} C^s_{ij'} W^{s'}_{j'}} ,$$

provided that the denominator is greater than zero.

Both for ease of preparing inputs and for consistency with other models that use this allocation method, the C_{ij}^{s}'s are not direct inputs to COMBAT. Instead (in algebraic notation), the inputs to COMBAT used here are:

$\hat{W}_{i}^{s}$ = the number of weapons of type i on side s in a typical-case force, where this force must contain a strictly positive number of weapons of each type being simulated, and where i = 1,N^s and s = 1,2.

$\hat{A}_{ij}^{s}$ = the average fraction of engagements that weapons of type i on side s would make against enemy weapons of type j (out of all of the engagements made by that type of weapon) when the enemy force consists of $\hat{W}_{j'}^{s'}$ weapons of type j', where i = 1,N^s, j = 1,$N^{s'}$, j' = 1,$N^{s'}$, and s = 1,2.

With these inputs, C_{ij}^{s} can be calculated as

$$C_{ij}^{s} = \hat{A}_{ij}^{s} / \hat{W}_{j}^{s'} .$$

Accordingly, for Method 0, the set $\bar{A}^{0}$ consists of $\hat{A}$ and $\hat{W}$ (as defined in Section B of Chapter II, the set $\bar{A}^{0}$ is used to help determine the allocations $A_{ij}^{s}(0)$). In particular, the allocation function F is rigorously specified by:

$$A_{ij}^{s}(0) = F_{ij}^{0s}(W,E,P,\bar{A}^{0})$$

$$= F_{ij}^{0s}(W,\hat{A},\hat{W})$$

$$= \begin{cases} \dfrac{\hat{A}_{ij}^{s} W_{j}^{s'} / \hat{W}_{j}^{s'}}{\sum_{j'=1}^{N^{s'}} \hat{A}_{ij'}^{s} W_{j'}^{s'} / \hat{W}_{j'}^{s'}} & \sum_{j'=1}^{N^{s'}} \hat{A}_{ij'}^{s} W_{j'}^{s'} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$= \begin{cases} \dfrac{C^s_{ij} W^{s'}_j}{\sum\limits_{j'=1}^{N^{s'}} C^s_{ij'} W^{s'}_{j'}} & \sum\limits_{j'=1}^{N^{s'}} C^s_{ij'} W^{s'}_{j'} > 0 \\ 0 & \text{otherwise}\,, \end{cases}$$

where C^s_{ij} is as defined above, and where $i = 1,N^s$, $j = 1,N^{s'}$, and $s = 1,2$. (Note that if a user of COMBAT wishes to input values for the C^s_{ij} directly, then this is easily done by setting $\hat{W}^{s'}_j = 1$ for all j.)

When aircraft (i.e., non-vulnerable weapons systems) are being simulated, COMBAT also accepts the input

$\hat{\underline{A}}^s_{ij}$ = the average fraction of engagements that aircraft of type i on side s would make against enemy weapons of type j (out of all of the engagements made by that type of aircraft) when the enemy force consists of $\hat{W}^{s'}_{j'}$ weapons of type j', where $i = 1,\underline{N}^s$, $j = 1,N^{s'}$, $j' = 1,N^{s'}$, and $s = 1,2$.

The allocation of fire for non-vulnerable weapons systems according to Method 0, $\underline{A}^s_{ij}(0)$, is then calculated using the same function as given above for $A^s_{ij}(0)$, except that $\hat{\underline{A}}^s_{ij}$ and $\underline{N}^s$ are used in place of $\hat{A}^s_{ij}$ and N^s.

This method for determining allocations of fire is used in IDAGAM, INBATIM, TACWAR, JCS FPM, and IDAPLAN, all of which are dynamic combat models. Discussions of various aspects of this method can be found in Chapter II of [8], on pages 98 through 100 of [9], on pages 31 and 32 of [10], on pages 53 and 54 of [11] (see also pages 42 and 43 of [11]), and on pages 4 through 8 of [12].

C. METHOD 1: ALLOCATIONS THAT ARE PROPORTIONAL TO PROBABILITIES OF KILL TIMES WEIGHTED NUMBERS OF TARGETS

A drawback of Method 0 is that it requires obtaining data for $\hat{A}$ (as defined above), yet it ignores data obtained for E and P (as defined in Section B of Chapter II) in determining the allocations of fire. The goals of Method 1 are: (1) to make reasonable use of data values for E and P, and (2) to allow but not require (for plausible use) that data be obtained for $\hat{A}$ (and $\hat{W}$).

There are many plausible ways to make use of E and P in determining allocations of fire. Reference [13] suggests considering the product

$$U_{ij}^{s} = E_{i}^{s} P_{ij}^{s} E_{j}^{s'} P_{ji}^{s'}$$

when determining how much fire weapons of type i on side s should allocate against enemy weapons of type j. Two ways to consider this product are as follows. First, U_{ij}^{s} could be used as a weighing factor in addition to, or instead of, C_{ij}^{s} within an approach analogous to Method 0. Second, U_{ij}^{s} could be used to determine pure priority allocations, which is the use discussed in [13]. This first approach is the basis for Method 1 as described here, the priority approach is used as part of Methods 2 and 3 as described below.

As indicated above, Method 1 allows data to be entered for $\hat{A}$ and $\hat{W}$. Accordingly, for Method 1, the set $\tilde{A}^{1}$ is the same as the set $\bar{A}^{0}$--the allocation function F is different. Specifically, for Method 1,

$$A_{ij}^{s}(1) = F_{ij}^{1s}(W,E,P,\tilde{A}^{1})$$

$$= \begin{cases} \dfrac{U_{ij}^{s}\,\hat{A}_{ij}^{s}\,W_{j}^{s'} / \hat{W}_{j}^{s'}}{\sum\limits_{j'=1}^{N^{s'}} U_{ij'}^{s}\,\hat{A}_{ij'}^{s}\,W_{j'}^{s'} / \hat{W}_{j'}^{s'}} & \sum\limits_{j'=1}^{N^{s'}} U_{ij'}^{s}\,\hat{A}_{ij'}^{s}\,W_{j'}^{s'} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$= \begin{cases} \dfrac{U^{s}_{ij}\, C^{s}_{ij}\, W^{s'}_{j}}{\sum\limits_{j'=1}^{N^{s'}} U^{s}_{ij'}\, C^{s}_{ij'}\, W^{s'}_{j'}} & \sum\limits_{j'=1}^{N^{s'}} U^{s}_{ij'}\, C^{s}_{ij'}\, W^{s'}_{j'} > 0 \\ 0 & \text{otherwise ,} \end{cases}$$

where $U^{s}_{ij} = E^{s}_{i}\, P^{s}_{ij}\, E^{s'}_{j}\, P^{s'}_{ji}$ and $C^{s}_{ij} = \hat{A}^{s}_{ij} / \hat{W}^{s'}_{j}$, and where $i = 1,N^{s}$, $j = 1,N^{s'}$, and $s = 1,2$.

. Note that, as a special case, data could be entered so that

$$\hat{W}^{s'}_{j} = 1 \qquad\qquad j = 1,N^{s'}$$

and

$$\hat{A}^{s}_{ij} = \hat{W}^{s'}_{j} \Big/ \sum_{j'=1}^{N^{s'}} \hat{W}^{s'}_{j'} = 1 / N^{s'} \qquad\qquad i = 1,N^{s},\ j = 1,N^{s'}$$

In this case, Method 1 becomes

$$A^{s}_{ij}(1) = \begin{cases} \dfrac{U^{s}_{ij}\, W^{s'}_{j}}{\sum\limits_{j'=1}^{N^{s'}} U^{s}_{ij'}\, W^{s'}_{j'}} & \sum\limits_{j=1}^{N^{s'}} U^{s}_{ij'}\, W^{s'}_{j'} > 0 \\ 0 & \text{otherwise .} \end{cases}$$

Thus, in this special case of Method 1, U^{s}_{ij} plays exactly the same role as C^{s}_{ij} plays in Method 0. Accordingly, the general form for F above allows E, P, $\hat{A}$, and $\hat{W}$ to all contribute to determining the allocations of fire; while the special case just described requires that no (substantive) data be obtained for $\hat{A}$ and $\hat{W}$, yet it produces a plausible allocation of fire that satisifes the criteria stated in Section A.1, above. (Conversely, if the special case

$$\hat{W}^{s}_{j} = 1 \text{ and } \hat{A}^{s}_{ij} = 1/N^{s'} \qquad\qquad i = 1,N^{s},\ j = 1,N^{s'}$$

is used in Method 0, then the allocation of fire becomes

$$A^s_{ij}(0) = W^{s'}_j \,/\, \sum_{j=1}^{N^{s'}} W^{s'}_j \,.$$

This allocation might be useful for theoretical purposes, but it does not satisfy the criteria stated in Section A.1 and, since it is independent of i, it is clearly unrealistic.)

As stated at the end of Section B, above, when aircraft (i.e., non-vulnerable weapons systems) are being simulated, COMBAT accepts the input $\hat{\underline{A}}$ as defined in that section. With one exception, the allocation of fire for aircraft using Method 1 is calculated using exactly analogous formulas as described above for fully interacting weapons systems--the exception concerns the calculation of U. For fully interacting weapons, U is defined by

$$U^s_{ij} = E^s_i P^s_{ij} E^{s'}_j P^{s'}_{ji} \qquad i = 1,N^s,\ j = 1,N^{s'},\ s = 1,2 \,.$$

For non-vulnerable weapons, define $\underline{U}$ by

$$\underline{U}^s_{ij} = \underline{E}^s_i \underline{P}^s_{ij} E^{s'}_j \sum_{i'=1}^{\underline{N}^s} P^{s'}_{ji'} \overline{W}^s_{i'} \qquad i = 1, \underline{N}^s,\ j = 1,N^{s'},\ s = 1,2 \,,$$

where

$$\overline{W}^s_i = W^s_i \,/\, \sum_{i'=1}^{\underline{N}^s} W^s_{i'} \qquad i = 1,N^s \,.$$

The allocation of fire for non-vulnerable weapons systems according to Method 1, $\underline{A}^s_{ij}(1)$, is then calculated using the same function as given above for $A^s_{ij}(1)$, except that $\underline{U}^s_{ij}$, $\hat{\underline{A}}^s_{ij}$, and $\underline{N}^s$ are used in place of U^s_{ij}, $\hat{A}^s_{ij}$, and N^s wherever they appear.

D. METHOD 2: ALLOCATIONS THAT ARE A CONVEX COMBINATION OF A PRIORITY BASED ON PK'S AND METHOD 1

While Method 1 calculates the products suggested in [13], it does not use these products in the way they are used in [13]. Method 1 uses these products as weighting factors, whereas [13] suggests that they be used as strict priorities. Method 2 starts with this suggestion and builds on it in the following manner.

Method 2 assumes that (on average) a fraction of the times that weapons of each type are engaging enemy weapons they can select the type of weapon to engage according to a strict priority, and one minus this fraction of times they will engage enemy weapons according to the weighted number of enemy weapons present, where the proportionality weights are determined using Method 1 as described above. Method 2 further assumes that this fraction can be an input that depends on the type of shooting weapon. That is, Method 2 requires the new input:

q_i^s = the average fraction of the engagements made by weapons of type i on side s that are made according to a strict priority, with one minus this fraction being made on a proportional basis, where $i = 1,N^s$ and $s = 1,2$.

Values for q_i^s must satisfy

$$0 \le q_i^s \le 1 \qquad\qquad i = 1,N^s \text{ and } s = 1,2 \,.$$

Since Method 2 needs values for q_i^s, and since (if $q_i^s < 1$) it needs values for $\hat{A}$ and $\hat{W}$, the set $\tilde{A}^2$ consists of $\hat{A}$, $\hat{W}$, and q (as defined in Section B of Chapter II, the set $\tilde{A}^2$ is used to help determine the allocations $A_{ij}^s(2)$).

To construct strict priorities based on values for U_{ij}^s as defined above, i.e.,

$$U_{ij}^s = E_i^s\, P_{ij}^s\, E_j^{s'}\, P_{ji}^{s'} \,,$$

define J_i^s as follows. For $i = 1,N^s$, let J_i^s be the set of j that maximizes U_{ij}^s over all j such that $1 \le j \le N^{s'}$ and $W_j^{s'} > 0$. To account for ties (i.e., cases in which J_i^s has more than one element) and to prevent "priority overkill," let

$$\Gamma_{ij}^s = \begin{cases} W_j^{s'} \Big/ \sum\limits_{j' \in J_i^s} W_{j'}^{s'} & j \in J_i^s \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1,N^s$ and $j = 1,N^{s'}$, and (if $\underline{N}^s = 0$) let

$$b_j^{s'} = \min\left\{1,\ W_j^{s'} \Big/ \sum_{i=1}^{N^s} q_i^s\, E_i^s\, \Gamma_{ij}^s\, P_{ij}^s\right\}$$

for $j = 1,N^{s'}$, where this minimum is taken to be one if the denominator of the ratio is zero. If $\underline{N}^s > 0$ then, as will be discussed below, $b_j^{s'}$ is calculated by

$$b_j^{s'} = \min\left\{1, W_j^{s'} \Big/ \left(\sum_{i=1}^{N^s} q_i^s E_i^s \Gamma_{ij}^s P_{ij}^s + \sum_{i=1}^{\underline{N}^s} \underline{q}_i^s \underline{E}_i^s \underline{\Gamma}_{ij}^s \underline{P}_{ij}^s \right)\right\}$$

where $\underline{\Gamma}$ is defined below. Let

$$\tilde{q}_{ij}^s = q_i^s \, b_j^{s'} \qquad i = 1,N^s \text{ and } j = 1,N^{s'} .$$

Then, for Method 2,

$$\begin{aligned} A_{ij}^s(2) &= F_{ij}^{2s}(W,E,P,\bar{A}^2) \\ &= \begin{cases} \tilde{q}_{ij}^s + (1-\tilde{q}_{ij}^s)\, A_{ij}^s(1) & j \in \Gamma_i^s \\ (1-\tilde{q}_{ij}^s)\, A_{ij}^s(1) & \text{otherwise ,} \end{cases} \end{aligned}$$

where $i = 1,N^s$, $j = 1,N^{s'}$, and $s = 1,2$.

Two special cases of this allocation worth noting are as follows. First, if the input q_i^s is given a value of zero, then $A_{ij}^s(2) = A_{ij}^s(1)$ for all relevant j. Thus, if q is identically zero, Method 2 reduces to Method 1. Second, setting q_i^s equal to one produces a special case (provided that "overkill" is not occurring) in which the third property listed in Section A.1 above is not satisfied. That is, setting q_i^s equal to one can result in a (special case) strict priority allocation as defined in that section.

Method 2 computes allocation of fire for non-vulnerable weapons (if any) using formulas that are directly analogous to those given above for fully interacting weapons systems. In particular, if non-vulnerable weapons systems are being simulated, then Method 2 also requires the input:

$\underline{q}_i^s$ = the average fraction of engagements made by non-vulnerable weapons of type i on side s that are made according to a strict priority, with one minus this fraction being made on a proportional basis, where $i = 1,\underline{N}^s$ and $s = 1,2$.

As with q_i^s, values for $\underline{q}_i^s$ must satisfy

$$0 \le \underline{q}_i^s \le 1 \qquad i = i,\underline{N}^s \text{ and } s = 1,2 .$$

For $i = 1,\underline{N}^s$, let $\underline{\Gamma}_i^s$ be the set of j that maximize $\underline{U}_{ij}^s$ over all j such that $1 \le j \le N^{s'}$ and

$W_j^{s'} > 0$, where $\underline{U}_{ij}^s$ is defined as in Method 1 (see Section C above). To account for ties and to prevent priority overkill, let

$$\underline{\Gamma}_{ij}^s = \begin{cases} W_j^{s'} \big/ \sum\limits_{j' \in \underline{I}_i^s} W_{j'}^{s'} & j \in \underline{I}_i^s \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1,\underline{N}^s$ and $j = 1,N^{s'}$, and (as stated above) let

$$b_j^{s'} = \min\left\{1,\ W_j^{s'} \Big/ \left(\sum_{i=1}^{N^s} q_i^s\, \Xi_i^s\, \Gamma_{ij}^s\, P_{ij}^s + \sum_{i=1}^{\underline{N}^s} \underline{q}_i^s\, \underline{E}_i^s\, \underline{\Gamma}_{ij}^s\, \underline{P}_{ij}^s \right)\right\} .$$

Let

$$\tilde{\underline{q}}_{ij}^s = \underline{q}_i^s \ b_j^{s'} \qquad i = 1,\underline{N}^s \text{ and } j = 1,N^{s'} .$$

Then, for Method 2,

$$\underline{A}_{ij}^s(2) = \begin{cases} \tilde{\underline{q}}_{ij}^s + (1 - \tilde{\underline{q}}_{ij}^s)\, \underline{A}_{ij}^s(1) & j \in \underline{I}_i^s \\ (1 - \tilde{\underline{q}}_{ij}^s)\, \underline{A}_{ij}^s(1) & \text{otherwise ,} \end{cases}$$

where $i = 1,\underline{N}^s$, $j = 1,N^{s'}$, and $s = 1,2$.

E. METHOD 3: ALLOCATIONS THAT ARE A CONVEX COMBINATION OF A PRIORITY BASED ON PK'S AND METHOD 0

The only difference between Method 2 and Method 3 is that Method 2 is a convex combination involving Pk's and Method 1 while Method 3 is a convex combination involving (in exactly the same way) Pk's and Method 0. In particular, for Method 2,

$$A_{ij}^s(2) = \begin{cases} \tilde{q}_{ij}^s + (1 - \tilde{q}_{ij}^s)\, A_{ij}^s(1) & j \in I_i^s \\ (1 - \tilde{q}_{ij}^s)\, A_{ij}^s(1) & \text{otherwise ,} \end{cases}$$

while for Method 3

$$A^s_{ij}(3) = \begin{cases} \tilde{q}^s_{ij} + (1 - \tilde{q}^s_{ij})\, A^s_{ij}(0) & j \in J^s_i \\ (1 - \tilde{q}^s_{ij})\, A^s_{ij}(0) & \text{otherwise ,} \end{cases}$$

where $\tilde{q}^s_{ij}$ and J^s_i are the same for both methods, and where $i = 1,N^s$, $j = 1,N^{s'}$, and $s = 1,2$.

Similarly, if non-vulnerable weapons are being simulated (i.e., $\underline{N}^s > 0$), then, for Method 2,

$$\underline{A}^s_{ij}(2) = \begin{cases} \underline{\tilde{q}}^s_{ij} + (1 - \underline{\tilde{q}}^s_{ij})\, \underline{A}^s_{ij}(1) & j \in \underline{J}^s_i \\ (1 - \underline{\tilde{q}}^s_{ij})\, \underline{A}^s_{ij}(1) & \text{otherwise ,} \end{cases}$$

while for Method 3

$$\underline{A}^s_{ij}(3) = \begin{cases} \underline{\tilde{q}}^s_{ij} + (1 - \underline{\tilde{q}}^s_{ij})\, \underline{A}^s_{ij}(0) & j \in \underline{J}^s_i \\ (1 - \underline{\tilde{q}}^s_{ij})\, \underline{A}^s_{ij}(0) & \text{otherwise ,} \end{cases}$$

where $\underline{\tilde{q}}^s_{ij}$ and $\underline{J}^s_i$ are the same for both methods, and where $i = 1,\underline{N}^s$, $j = 1,N^{s'}$, and $s = 1,2$.

F. METHOD 4: ALLOCATIONS THAT ARE DETERMINED BY GROUP DETECTIONS AND RIGID PRIORITIES BASED ON PK'S

As noted above, this method for determining allocations of fire has not been coded, and so it is not currently available for use in COMBAT. The reasons for discussing this method here are twofold. First, and more importantly, this method is discussed in order to help put in context the allocation methods and the properties these methods must satisfy as described above. Second, this method is described so that, if desired, it can be incorporated into COMBAT or into other models.

The basic ideas behind Method 4 are the following assumptions: (1) a (shooting) weapon can engage an enemy weapon only if it detects that enemy weapon. (2) The number of enemy weapons that a shooting weapon detects depends on the type of shooting weapon but is independent of the number and types of enemy weapons present (provided only that there are sufficiently many enemy weapons present). (3) If a total of $\bar{W}$ enemy weapons are present, if a total of t of these enemy weapons are detected by a shooting weapon, and if W_j of these $\bar{W}$ enemy weapons present are weapons of type j, then the probability that exactly x enemy weapons of type j are detected by that shooting weapon is assumed to be given by $\binom{W_j}{x}\binom{\tilde{W}-W_j}{t-x} + \binom{\tilde{W}}{t}$, where $\binom{x}{r} \equiv x(x-1)\ldots(x-r+1)/r!$ if $r \geq 1$ and $\binom{x}{r} \equiv 1$ if $r = 0$. (Thus, this detection process can be viewed as being similar to a sampling without replacement scheme.) (4) Each shooting weapon is assumed to select one enemy weapon to attack from among the highest priority type of enemy weapons it has detected, where the priority order used for shooting weapons of type i on side s is strictly based on U^s_{ij} as described below.

Unlike Methods 0, 1, 2, and 3, Method 4 does not use the inputs $\hat{A}$ or $\hat{W}$. Instead, to implement the assumptions just stated, Method 4 uses the inputs T^s_i and $\bar{Q}^s_i(t)$ defined as follows:

T^s_i = the maximum number of enemy weapons that a weapon of type i on side s can effectively detect in one time period, where $i = 1,N^s$ and $s = 1,2$.

$\bar{Q}^s_i(t)$ = the probability that a weapon of type i on side s detects exactly t enemy weapons in one time period given that there are at least t enemy weapons present, where $t = 1,T^s_i$, $i = 1,N^s$, and $s = 1,2$.

Thus, for Method 4, $\tilde{A}^4 = \{T, \bar{Q}\}$ for T and $\bar{Q}$ as just defined (as stated in Section B of Chapter II, the set $\tilde{A}^4$ is used to help determine the allocations $A^s_{ij}(4)$).

As in Methods 1, 2, and 3, let

$$U^s_{ij} = E^s_i P^s_{ij} E^{s'}_j P^{s'}_{ji}$$

for $i = 1,N^s$, $j = 1,N^{s'}$, and $s = 1,2$. For Method 4 only, let

$$J^s_i(1),\ J^s_i(2),\ \ldots,\ J^s_i(N^{s'})$$

be such that

$$U^s_{iJ^s_i(1)} \geq U^s_{iJ^s_i(2)} \geq \ldots \geq U^s_{iJ^s_i(N^{s'})},$$

with ties being considered in some suitable manner. This definition is structured so that $J^s_i(\iota) = j$ means that enemy weapons of type j are the ι^{th} priority weapons for attack by weapons of type i on side s.

For s = 1,2 let

$$\dot{W}_1^{s'} = \sum_{j=1}^{N^{s'}} W_j^{s'} ,$$

$$\dot{W}_2^{s'}(i) = \dot{W}_1^{s'} - W_{J_i^s(1)}^{s'} , \qquad i = 1,N^s ,$$

$$\dot{W}_3^{s'}(i) = \dot{W}_1^{s'} - W_{J_i^s(1)}^{s'} - W_{J_i^s(2)}^{s'} , \qquad i = 1,N^s , \dots ,$$

$$\dot{W}_{N^{s'}}^{s'}(i) = W_{J_i^s(N^{s'})}^{s'} , \qquad i = 1,N^s, \text{ and}$$

$$Q_i^s(t) = \begin{cases} \bar{Q}_i^s(t) & t < \dot{W}_1^{s'} , \\ \sum_{t'=t}^{T_i^s} \bar{Q}_i^s(t') & t = \dot{W}_1^{s'} , \\ 0 & t > \dot{W}_1^{s'} , \qquad i = 1,N^s . \end{cases}$$

Then an allocation of fire satisfying the assumptions stated above can be constructed using the following formulas.

$$B_{it}^{s1} = \begin{cases} 1 & \dot{W}_2^{s'}(i) < t \\ \left\{ 1 - \left[\binom{\dot{W}_2^{s'}(i)}{t} \div \binom{\dot{W}_1^{s'}}{t} \right] \right\} & \dot{W}_2^{s'}(i) \geq t , \end{cases}$$

$$A_{U_i^s(1)}^{s}(4) = \sum_{t=1}^{T_i^s} Q_i^s(t) B_{it}^{s1} \quad ,$$

$$B^{s2}_{it} = \begin{cases} 1 & t > \ddot{W}^{s}_{3}(i) \\ \left\{ 1 - \left[\binom{\ddot{W}^{s'}_{3}(i)}{t} \div \binom{\ddot{W}^{s'}_{2}(i)}{t} \right] \right\} & t \leq \ddot{W}^{s'}_{3}(i) , \end{cases}$$

$$A^{s}_{iJ^s_i(2)}(4) = \sum_{t=1}^{T^s_i} Q^s_i(t) \left(1 - B^{s1}_{it} \right) B^{s2}_{it} \quad ,$$

$$\vdots$$

$$A^{s}_{iJ^s_i(N^s)}(4) = 1 - A^{s}_{iJ^s_i(1)} - \cdots A^{s}_{iJ^s_i(N^s-1)} \; .$$

It should be relatively clear how to extend this notation, if desired, to explicitly consider non-vulnerable weapons as described above.

In the equations for A above, note that all of the combinatiorial terms are of the form

$$\binom{w-x}{t} \div \binom{w}{t} .$$

To calculate an exact value for this ratio, let

$$u = \max\{t,x\}$$

and

$$v = \min\{t,x\} .$$

Then

$$\binom{w-x}{t} \div \binom{w}{t} = \frac{w-u}{w} \times \frac{w-u-1}{w-1} \times \frac{w-u-2}{w-2} \times \ldots \times \frac{w-u-(v-2)}{w-(v-2)} \times \frac{w-u-(v-1)}{w-(v-1)} .$$

An approximate value for this ratio can be calculated using Forsyth's formula:

$$n! \cong \sqrt{2\pi}\left(\frac{\sqrt{n^2+n+1/6}}{e}\right)^{n+1/2},$$

which gives tha.

$$\binom{w-x}{t} + \binom{w}{t} \cong$$

$$\left[\frac{\left[\dfrac{(w-v)^2+(w-v)+1/6}{w^2+w+1/6}\right]^{((w-v)/2+1/4)}}{[w^2+w+1/6]^{(v/2)}}\right] + \left[\frac{\left[\dfrac{(w-u)^2+(w-u)+1/6}{(w-z)^2+(w-z)+1/6}\right]^{((w-u)/2+1/4)}}{[(w-z)^2+(w-z)+1/6]^{(u/2)}}\right],$$

where u and v are as defined just above and z = x + t.

A characteristic of this allocation method that may be of interest is as follows. Unlike Methods 1, 2, and 3 above, but like the allocation methods described in [6] and [7], this method uses priorities in a rigid manner. For example, if a shooting weapon of type i has detected exactly two enemy weapons, one of type j and one of type j', and if (according to J_i^S) weapons of type j have a higher priority than weapons of type j', then that shooting weapon always fires at the enemy weapon of type j--it never attempts to engage the enemy weapon of type j' instead. However, like Methods 1, 2, and 3 above, but unlike the allocation methods described in [6] and [7], this method satisfies the third property stated in Section A.1 above.

The essential difference between this method and those of [6] and [7] is as follows. Using this method, the probability that a shooting weapon detects (and hence engages) a high priority target is smaller if it is facing (say) 20 high priority targets and 2000 low priority targets than it is if it is facing 20 targets of each priority. Conversely, in [6] and [7] the probability that a shooting weapon detects (and hence engages) a high priority target remains constant when it is facing (say) 20 high priority targets no matter how many targets of lower priority it is also firing. This distinction can be quite important in dynamic models because the relative numbers of weapons can change (frequently significantly) over the course of the combat being simulated.

IV. ALLOCATIONS OF FIRE THAT DEPEND ON WEAPONS SCORES

A. STRUCTURE

Given a set of weapons scores (i.e., given V^s_i for $i = 1,N^s$ and $s = 1,2$), it is a relatively straightforward task to modify Methods 1 through 4 of Chapter III in order to produce allocations of fire that depend on those weapons scores. In particular, Methods 1 through 4 of Chapter III all make use of the product

$$U^s_{ij} = E^s_i P^s_{ij} E^{s'}_j P^{s'}_{ji}$$

in order to determine a priority of fire for weapons of type i on side s. Thus, in a sense, these priorities assume that value of killing a weapon of type j is given by $E^{s'}_j P^{s'}_{ji}$ for shooting weapons of type i, and that priorities are set in order to maximize the resulting value killed. Conversely, it can be argued that: (1) the value of killing a weapon should be indpendent of the type of shooter doing the killing, (2) weapons scores (computed either as described in Chapter II or by some other means) can serve as reasonable (shooter-independent) values for killing enemy weapons, and so (3) priorities should be set to maximize value killed where the value of killing an enemy weapon is the score of that weapon. Thus, in place of using the product

$$U^s_{ij} = E^s_i P^s_{ij} E^{s'}_j P^{s'}_{ji}$$

to construct priorities, according to this argument the product

$$\tilde{U}^s_{ij} = E^s_i P^s_{ij} V^{s'}_j$$

should be used instead.

Relationships concerning input weapons scores, calculated weapons scores, and dynamic uses of the model are somewhat more complex. In particular, five general approaches for structuring these relationships are as follows.

First, fixed weapons scores could be obtained by an external method (i.e., a method other than one of those described in Chapter II) that does not depend on having first computed an allocation of fire. For example, such scores could be determined by setting the score of a weapon equal to an estimate of the cost of producing that weapon. As another example, a commonly used set of scores is obtained by setting the score of any armored vehicle equal to one and the score of all other types of weapons equal to zero. Since such fixed scores do not depend on allocations of fire, they can be set by direct input and used to compute $\tilde{U}^{s}_{ij}$ as defined above, and then these $\tilde{U}^{s}_{ij}$ can be used to determine allocations of fire by methods directly analogous to Methods 1 through 4 of Chapter III.

Second, fixed weapons scores could be obtained using any one of the methods discussed in Chapter II by using that method in conjunction with a fixed but typical allocation of fire. For example, one of the scoring methods described in Chapter II could be selected. Then, for each side, a typical allocation of fire against a typical opposing force could be constructed, and a set of weapons scores based on that method and that allocation of fire could be calculated in a "one-time" run of COMBAT. The resulting weapons scores could then be used as input values for setting weapons scores by direct input. COMBAT (or another dynamic model) could then be run using these fixed (input) scores to determine allocations of fire based on $\tilde{U}^{s}_{ij}$ as described above. Since these allocations of fire depend on the numbers of weapons in addition to these weapons scores, the allocations of fire would change over time even though the weapons scores remain fixed.

Third, suppose it is desired to analyze a defense issue by running a set of cases using COMBAT (or another dynamic model), and suppose one of those cases can be designated as a reasonably comprehensive base case. Then weapons scores and allocations of fire for the first time period of that base case could be computed by solving a set of simultaneous equations, and then the resulting weapons scores could be used as fixed scores (set through direct input) for all time periods of all of the cases being run.

The distinction between this third general approach and the second one described above is as follows. The second approach uses typical allocations that are independent of weapons scores to compute a set of typical scores based on these typical allocations. It then computes all allocations (including the allocations for the first time period of a base case) based on these typical scores. Thus, the allocation of fire for the first time period of a base case will not (in general) be the same as the typical allocation of fire. However, this second general approach does not require solving simultaneous equations for both scores and allocations. Conversely, the third general approach assumes that the first time period base-case scores are a function of the first time period base-case allocation and vice versa, so that a set of simultaneous equations must be solved to yield these scores and allocations. The resulting weapons scores are then held fixed, and all scores for all time periods for all of the cases being considered are set equal to these fixed scores by direct input. Accordingly, unlike the second approach, the weapons scores for the first time period of the base case will be those that result from the allocations of fire used for (the first time period of) that base case.

The fourth general approach here is to solve a set of simultaneous equations to determine the weapons scores and allocations of fire for the first time period of each run of COMBAT (or of a similar dynamic model), but after the first time period only the allocations of fire change--the weapons scores remain fixed as if they were set by direct input. (Note that, in all of these approaches, the allocations of fire can always change each time period, even if weapons scores are held fixed, because the numbers of weapons involved can change between time periods.)

The distinction between the third and fourth approaches is as follows. In doing a series of runs, the third approach requires designating one run as a base case, and it solves a set of simultaneous equations for weapons scores and allocations of fire only for the first time period of that base case. Conversely, the fourth approach does not require designating any particular run as a base case, but it must solve a set of simultaneous equations for weapons scores and allocations of fire for the first time period of each case being run.

The fifth general approach here is to solve a set of simultaneous equations to determine weapons scores and allocations of fire for each time period of each case being run.

Table IV-1 summarizes selected characteristics of these five general approaches.

B. METHODS TO ALLOCATE FIRE THAT DEPEND ON WEAPONS SCORES

Repeating the notation introduced above, let V_i^s denote the score being used for weapons of type i on side s, where $i = 1,N^s$ and $s = 1,2$, and let

$$\tilde{U}_{ij}^s = E_i^s P_{ij}^s V_j^{s'}$$

for $i = 1,N^s$, $j = 1,N^{s'}$, and $s = 1,2$. In addition, let

Table IV-1. A Summary of Selected Characteristics Concerning Five General Approaches for Relating Allocations of Fire and Weapons Scores

General Approach	Source of Method Used to Determine Weapons Scores	Number of Solutions of Simultaneous Equations for Scores and Allocations that are Required	Variability of Weapons Scores Over Time Periods and Runs	Is a Set of Typical Allocations of Fire Required	the Designation of a Base Case Required
1	External	Zero	Scores remain fixed throughout	Perhaps, depending on external method	No
2	Internal	Zero	Scores remain fixed throughout	Yes	No
3	Internal	One per set of runs	Scores remain fixed throughout	No	Yes
4	Internal	One per run	Scores can vary over runs but not over time periods within a run	No	No
5	Internal	One for each time period of each run	Scores can vary both over runs and over time periods within a run	No	No

$$\underline{\tilde{U}}_{ij}^{s} = \underline{E}_i^s \underline{P}_{ij}^s \underline{V}_j^{s'}$$

for $i = 1,\underline{N}^s$, $j = 1,N^{s'}$, and $s = 1,2$. With this notation, formulas for Methods 1 through 4 here are directly obtained from the formulas for Methods 1 through 4 of Chapter III by replacing U and $\underline{U}$ with $\tilde{U}$ and $\underline{\tilde{U}}$, respectively, wherever U and $\underline{U}$ appear in Chapter III. With this replacement of U and $\underline{U}$ by $\tilde{U}$ and $\underline{\tilde{U}}$, Method 1 gives allocations that are proportional to value killed times weighted numbers of targets, Method 2 gives allocations that are a convex combination of a priority based on value killed and Method 1, Method 3 gives allocations that are a convex combination of a priority based on value killed and Method 0, and Method 4 gives allocations that are determined by group directions and rigid priorities based on value killed.

V. ATTRITION EQUATIONS FOR AIMED FIRE

A. STRUCTURE

1. General Approaches for Modeling Attrition

Attrition can be calculated in models of combat using four general approaches. First, in a few cases (such as a nuclear attack on undefended soft targets) it can be reasonable to assume that attrition is essentially deterministic, and so appropriate deterministic methods can be used to calculate attrition in these cases. Second, in a few cases the spectrum of combat being considered is simple enough that stochastic results (such as probability distributions or expected values of resulting random variables) can be rigorously computed. Third, stochastic results can be estimated (but not rigorously computed) using basically deterministic methods (this approach will be discussed further, below). Fourth, Monte Carlo techniques can be employed.

COMBAT is not intended to model those (relatively rare) cases for which it is reasonable to assume that attrition is essentially deterministic, and so the first approach above does not apply. COMBAT can be used to properly compute expected values for certain very special cases; but since COMBAT is not limited to modeling only these very special cases, the second approach above does not (in general) apply. Thus, as in most other dynamic models of combat, the choice of approach to compute attrition reduces to choosing either the third or the fourth approach as noted above.

A thorough discussion of the advantages and disadvantages of these approaches is beyond the scope of this paper. The third approach was selected for the following three reasons. First, "deterministic estimation" models typically run much more quickly than multiple trials of Monte Carlo models (all other things being equal), and quick running time was desired here. Second, the advantages of the Monte Carlo approach over the deterministic estimation approach tend to be more significant for fine-grained models than for more highly aggregated models, and COMBAT is a relatively highly aggregated model. Third, and perhaps most important here, this choice allows COMBAT to be used as a vehicle for reporting some ongoing research on attrition equations, and the aggregated nature of COMBAT renders it quite appropriate for this use.

2. The Deterministic Estimation Approach for Modeling Attrition

The vast majority of deterministic estimation models, including COMBAT, use the following general algorithm to construct a deterministic surrogate for the stochastic process they are attempting to address. For this discussion only, consider the following notation. Suppose the model in question considers a total of m types of resources (on both sides) and a total of n possible interactions (e.g., time periods). Let the initial number of resources of type i be denoted by X_{i0} for i = 1,m. For i = 1,m and j = 1,n, let X_{ij} be the random number of resources of type i after the jth interaction. If resources of type i are not included in the first interaction, then $X_{i1} = X_{i0}$ with probability one. However, if resources of type i are involved in the first interaction, then X_{i1} is generally a non-degenerate random variable. For j = 0,n, let $Y_j = \{X_{1j},...,X_{mj}\}$, and let f_j denote the jth interaction in that

$$Y_j = \{X_{1j},...,X_{mj}\} = f_j(\{X_{1,j-1},...,X_{m,j-1}\}) = f_j(Y_{j-1})$$

Given this notation, the desired outputs of the model are the expected values of the random variables $X_{1n},...,X_{mn}$. That is, the goal here is to estimate values for

$$\begin{aligned} E[Y_n] &= E[\{X_{1n},...,X_{mn}\}] \\ &= E[f_n(f_{n-1}...f_1(\{X_{10},...,X_{m0}\})...)] \\ &= E[f_n...f_1(Y_0)] \ . \end{aligned}$$

COMBAT, and most other deterministic estimation models, estimate these expected values in the following manner. First, expected values of $X_{11},...,X_{m1}$ are either rigorously computed or are estimated using bounded approximations or reasonable heuristics. Let $\overline{X}_{11},...,\overline{X}_{m1}$ denote (perhaps an estimate of) these expected values. Thus, $\overline{X}_{11},...,\overline{X}_{m1}$ are deterministic quantities, not random variables. The model then computes $\dot{X}_{12}$ through $\dot{X}_{m2}$ as:

$$\{\dot{X}_{12},...,\dot{X}_{m2}\} = f_2(\{\overline{X}_{11},...,\overline{X}_{m1}\})$$

or, in terms of the Y's,

$$\breve{Y}_2 = f_2(\bar{Y}_1)$$

Note that, while $\bar{Y}_1$ may be a rigorously computed expectation of the random quantity $f(Y_0)$, the same is not true of $\breve{Y}_2$ since (in general)

$$E[Y_2] = E[f_2(Y_1)] \neq f_2(E[Y_1]) .$$

This replacement of the expectation of a function by the function of the expectation is made for all succeeding interactions. That is, let $\bar{E}_j$ denote the estimation technique being used to compute individual estimated values for the j^th^ interaction (e.g., if each expectation for each individual interaction is being computed rigorously, then $\bar{E}_j[\bullet] = E[\bullet]$ for all j). Then

$$\begin{aligned}\breve{Y}_n &= \bar{E}_n[f_n(\breve{Y}_{n-1})] \\ &= \bar{E}_r[f_n(\bar{E}_{n-1}[f_{n-1} \ldots f_2(\bar{E}_1[f_1(Y_0)]) \ldots])] .\end{aligned}$$

Like other deterministic estimation models, COMBAT does not attempt to estimate the difference between $\breve{Y}_n$ and $E[Y_n]$, where $\breve{Y}_n$ is as just defined and $E[Y_n]$ is properly calculated as

$$E[Y_n] = E[f_n \ldots f_1(Y_0)] .$$

Clearly, hypothetical cases can be constructed in which this difference is quite large. In realistic cases, this difference might be small, might be somewhat large but relatively unimportant, or might be large and quite significant, depending on the data being used and the issue being addressed.

3. General Approaches for Processing Time

Time can be processed in dynamic models of combat using four general approaches. First, a model can have the property that it continuously simulates the passing of time (perhaps, on a particular computer, at a speed faster than, or equal to, or slower than the passage of real time, or perhaps at varying speeds). Second, a model can step through time in steps of fixed or independently-determined size--such a model is frequently called a time-step model. In particular, in a time-step model time is advanced by a fixed or independently-determined amount to a new point in time, and the states or statuses of resources are updated as of that new point in time. Typically these time steps are of constant size, but they need not be. Third, a model can build a list of significant (to it) events and, after it simulates one event, it steps directly to the time of the next event, no matter how long or how short that step in time is. Such a model is frequently called an event-step (or event-store) model. In particular, in an event-step model, selected events are scheduled in time, time is advanced to the occurrence of the next scheduled event, and the states or statuses of resources (as well as the schedule of upcoming events) are updated at that point in time to reflect the occurrence of that event. Finally, a dynamic model can be simple enough that it has a closed form solution in that the states or statuses of resources are described as explicit and computationally tractable functions of time. In this case, the status of a resource at, say, time *t* can be found by evaluating the appropriate function at *t*, without having to simulate combat (either in steps or continuously, as described above) from the start of that combat through time *t*.

Very few models of combat are simple enough to have a closed form solution, and COMBAT is not such a simplistic model.

The distinction between time-step and event-step models is primarily useful for Monte Carlo models. Event-step deterministic estimation models are relatively rare and the comments below, reworded slightly, would apply for all practical purposes to such models. Also, there is essentially no practical difference between a continuously running model and a time-step model with very short time periods. Accordingly, it is useful to picture deterministic estimation models, like COMBAT, as being time-step models which may be run using relatively short, intermediate, or relatively long time periods.

4. The Time-Step Approach with Relatively Long Time Periods for Processing Time

In an abstract sense, the distinction between using short time periods versus using long time periods in a time-step model is judgemental and relative. In a practical sense, however, this distinction tends to be relatively clear. In a few cases, time-step models are used to approximate continuous time in that the time periods are selected to be so short that the probability that two or more changes in the state of resources occur during one time period is practically zero. In these cases, if estimation of the results show that more that one such change is likely to happen in one time period, then the length of the time period should be reduced. In many cases, however, time periods are set to be sufficiently long that several days of combat can be simulated in a comparable number of time steps--e.g., a time period may be between a quarter of a day and four days long. Clearly many resources can change states during time periods of this length.

This distinction between short and long time periods is particularly important for the following reason. Early models of attrition (i.e., Lanchester equations) were originally formulated as differential equations, implying very short time periods when coded in time-step form. Whether or not such models are appropriate as differential equations, they may be quite inappropriate for use as difference equations with long time periods. Conversely, other models of attrition might produce quite similar results (compared to Lanchester equations) when coded as differential equations, but might produce quite different results when coded as difference equations with long time periods.

For example, using the notation of Chapter II, the Lanchester square differential equation can be written as

$$\frac{dW_j^{s'}(t)}{dt} = \begin{cases} -\sum_{i=1}^{N^s} K_{ij}^s W_i^s(t) & W_j^{s'}(t) > 0 \\ 0 & \text{otherwise} \end{cases}$$

for $j = 1, N^{s'}$ and $s = 1,2$. In the homogeneous case ($N^1 = N^2 = 1$), this equation becomes

$$\frac{dW^{s'}(t)}{dt} = \begin{cases} -K^s W^s(t) & W^{s'}(t) > 0 \\ 0 & \text{otherwise}\,. \end{cases}$$

for $s = 1,2$. A difference equation version of this homogeneous relationship is

$$\Delta W^{s'} = -\min\{K^s W^s, W^{s'}\}$$

with the definition of K being adjusted to compensate for any change in time scale. Note, however, that this latter relationship allows multiple kills to occur in one time period, which can cause problems as the following numeric example shows.

Let $N^1 = N^2 = 1$, and let $E^1 = 1$, $P^1 = 2/3$, $W^1 = 3$, and $W^2 = 2$. Thus, $K^1 = E^1 P^1 = 2/3$ and

$$\Delta W^2 = -\min\{(2/3)3, 2\} = -2\,.$$

Conversely, it can be argued that the theoretical best that side 1 could possibly achieve with three shooters is to have a shoot-look-shoot-look-shoot capability, which results in

$$\Delta W^2 = -46/27\,.$$

Thus, use of Lanchester square in this case overstates the theoretical maximum number of kills that side 1 could achieve by 17% in that

$$2/(46/27) = 1.17.$$

If, instead of having such a shoot-look-shoot capability, each of the three shooters on side 1 randomly (i.e., uniformly and independently) selects a target to engage, then the expected number of kills drops from 46/27 to 38/27, and so the corresponding overstatement caused by using the Lanchester equation grows to 42% in that

$$2/(38/27) = 1.42.$$

The point here is certainly not that this example is a relatively realistic portrayal of typical combat results. Instead, the point is as follows. First, it frequently has been (and likely will continue to be) appropriate to use deterministic estimation time-step combat models with relatively large time periods. Second, attrition calculations in such models can vary significantly when using simplistic difference equation extensions of differential equations as compared to using directly derived difference equations. Third, attrition calculations can also differ significantly when using various directly derived difference equations--in the example above:

$$(46/27)/(38/27) = 1.21,$$

a difference of 21%.

5. Implications and Resulting Structure

Based in part on these arguments, the general structure used to calculate attrition in COMBAT is as follows.

COMBAT is designed to allow (though, of course, not necessarily require) relatively long time periods. For example, each time step in COMBAT might correspond to a time period of between one quarter day and four days long. This aspect of COMBAT means that many events (i.e., losses of weapons due to enemy fire) can occur during each time period. Two major implications of this aspect of COMBAT are as follows. First, some of the weapons that were operational at the start of a time period, and could have fired lethal shots during the time period, should not be able to do so because they are destroyed before they can fire during that time period The attrition calculations in COMBAT are structured to account for this implication of relatively long time periods. Second, occasionally two or more weapons on one side will fire potentially lethal shots at the same target on the other side, which (if there are s shooters with a probability of kill of k) would not, in general, result in an overall probability of killing the target greater than $1 - (1-k)^s$. In particular, if two shooters with a probability of kill of 0.5 shoot at one target, and if the effects of those two shots are independent, then the probability that the target is killed is 0.75, not $2 \times (0.5) = 1.0$ as would be implied by a simplistic Lanchester square difference equation. Use of a Lanchester square difference equation is included as an option (for comparison purposes) in COMBAT. However, except for this option (i.e., if any of the alternative options for calculating unilateral attrition is selected instead), COMBAT is structured to prevent such overkilling from occurring.

Some attrition structures exist that can simultaneously consider both of these implications of relatively long time periods--see, for example, Reference [14]. However, as in [14], these attrition structures tend to be appropriate only in special cases, and COMBAT is not designed to be limited to these special cases. Such special attrition structures could be added to COMBAT, if desired, to model special cases; but, in order to address more general cases, COMBAT needs and uses a more general structure to address these two implications of relatively long time periods. (Note that both of these implications, if not addressed, would result in overestimating the numbers of weapons killed.) The general attrition structure used in COMBAT is as follows.

The attrition calculations in COMBAT can be considered as consisting of two segments. In the first segment, four unilateral attrition assessments are calculated. Each of these attrition assessments is unilateral in that, for the assessment in question, only one of the two sides is firing at the other and so only the other side is suffering attrition. In two of these assessments side 1 is firing at side 2 (the distinction between these two assessments is explained in Section C), and in the other two side 2 is firing at side 1. The second segment combines the results of these four assessments to obtain the overall attrition to both sides. Each segment allows a choice among various options for the calculations made by that segment. The options currently available for calculating unilateral attrition are discussed in Section B, below, and the options available for calculating overall attrition from these four unilateral assessments are discussed in Section C.

B. UNILATERAL ATTRITION EQUATIONS

A definitive discussion of the distinctions between modeling aimed fire and modeling other types of fire is beyond the scope of this paper. However, several points should be noted. First, cases exist where these distinctions are not simple to make. For example, Reference [15] presents a model of attrition which is not easily categorized as being either a type of aimed fire or a type of area fire (it is, in a sense, a mixture of each). Second, an attrition equation suitable for modeling aimed fire might also, with a different definition of its parameters, be suitable for modeling certain types of area fire. That is, the functional form of the attrition equation would be the same for aimed fire as for certain types of area fire, but the assumptions and definitions of the inputs used would be different. Such alternative assumptions and definitions for area fire can be made for each of the unilateral attrition equations discussed below. Finally, assuming that some weapons engage the enemy using aimed fire need not exclude assuming that other weapons engage the enemy using other types of fire. COMBAT currently does not allow such a structure, but it could relatively easily be modified to do so. In particular, one option currently available in COMBAT allows a mixture of two different types of aimed fire (this option is discussed below), and this concept could be extended to allow a mixture of aimed and area fire. Such an extension would (in essence) require adding one or more types of area fire attrition equations to the model. Recent work on equations suitable for modeling relatively general types of area fire is discussed in Reference [16].

Except for this section, the notation required in this paper is necessarily two-sided. However, since this section considers only unilateral attrition equations (that is, for these equations, one side is firing at the other but not vice versa), two-sided notation is not needed. To simplify the presentation that follows (at no loss in generality), Subsection 1 below gives some one-sided notation that will be used in (but only in) the remaining subsections of this section. Subsection 2 presents an overview of the remaining subsections, and Subsections 3 through 7 discuss various unilateral attrition equations for modeling aimed fire.

1. Notation for Unilateral Attrition

Consider the following (one-sided) notation.

m = the number of types of weapons on the shooting side.

n = the number of types of weapons on the target side.

s_i = the number of engagements that can be made by all weapons of type i on the shooting side, where $i = 1,m$.

t_j = the number of vulnerable weapons of type j on the target side, where $j = 1,n$.

a_{ij} = $a_{ij}(t_1,\ldots,t_n)$ = the average fraction of engagements (i.e., the allocation of fire) that weapons of type i on the shooting side make against enemy weapons of type j (out of all of the engagements made by those type-i weapons) when the target side consists of $t_{j'}$ vulnerable weapons of type j', where i = 1,m, j = 1,n, and j' = 1,n.

p_{ij} = the probability of kill per engagement by a weapon of type i on the shooting side when that weapon is engaging an enemy weapon of type j, where i = 1,m and j = 1,n.

COMBAT is coded to allow only an input fraction of weapons to be vulnerable to enemy fire. In terms of the (two-sided) notation of Chapter II, Section B, this input fraction is defined as

U_i^s = the fraction of weapons of type i on side s that are vulnerable to enemy fire, where i = 1,N^s and s = 1,2.

COMBAT is also coded to allow partitioning the weapons on both sides into an input number of identical combat areas. If C denotes this number of combat areas (and C is greater than zero), then attrition is calculated by assuming that 1/C of the weapons on each side are associated with each of C combat areas and that the overall attrition is the product of C times the number of weapons killed in one combat area. (The code treats C = 0 as if C = 1.)

With this additional notation, the one-sided notation introduced here can be related to the two-sided notation of Chapter II, Section B, for side s shooting at side s' (s' = 3–s) by:

$m = N^s$

$n = N^{s'}$

$s_i = E_i^s W_i^s / C$ i = 1,m

$t_j = U_j^{s'} W_j^{s'} / C$ j = 1,n

$a_{ij} = A_{ij}^s$ i = 1,m and j = 1,n, and

$p_{ij} = P_{ij}^s$ i = 1,m and j = 1,n .

The unilateral attrition equations discussed below compute a number of target weapons of type j killed per combat area, Δt_j, as a function of m, n, s, t, a, and p.

In the remainder of this chapter the term "shooter" will be used in place of "engagement by a shooting weapon." For example, the number of engagements by shooting weapons of type i (i.e., s_i) will be called the number of shooters of type i in the discussion below.

2. Overview of Options for Calculating Unilateral Attrition

COMBAT currently allows five different options for calculating unilateral attrition. Four of these options differ (primarily) in the degree of coordination that the shooting weapons are assumed to possess. The fifth option, Lanchester square, can be viewed either as an option included solely for comparison purposes or as an option that represents an (impossible to actually achieve) upper bound on the level of coordination among weapons. A brief overview of the other four levels of coordination is as follows.

The lowest level of coordination simulated in COMBAT uses a binomial type of attrition equation. The key coordination assumption behind this equation is that each shooter selects a target to shoot at independently of the selections made by other shooters. (Note that this implies that a weapon that can make two or more engagements selects its target for each of these engagements independently of its other selections as well as independently of the selections made by other shooting weapons.)

Skipping the second level of coordination for a moment, the third level of coordination simulated in COMBAT uses a uniform type of attrition equation. The key coordination assumption behind this equation is that the shooting side will make

$$\sum_{i=1}^{m} s_i a_{ij}$$

engagements against enemy weapons of type j and will distribute these engagements as uniformly as possible over the t_j targets of type j present.

The second level of coordination simulated in COMBAT requires an additional input. In terms of the (two-sided) notation of Chapter II, Section B, this input is defined as

Z_i^s = the fraction of shooting weapons of type i on side s that select targets independently (i.e., have the lowest level of coordination as described above) with the remainder $(1 - Z_i^s)$ of these shooting weapons distributing their fire uniformly over all targets of the type they are allocating this fire against, where $i = 1, N^s$ and $s = 1,2$.

This input is used only if the second level of coordination is selected. If this second level is selected then, in order to survive, targets must survive $z_i s_i$ uncoordinated engagements and $(1 - z_i)s_i$ uniformly distributed engagements, where $z_i = Z_i^s$.

The fourth (and highest physically possible) level of coordination simulated in COMBAT is a shoot-look-shoot attrition structure. This structure assumes that the shooters attack one-at-a-time, and each shooter knows the outcome of all previous engagements before it selects a target for its engagement. Knowledge of all previous outcomes means that a shooter never fires at a target that has already been killed by another shooter.

Section 3, below, discusses the uncoordinated (binomial) attrition option. Section 4 discusses the uniform-fire attrition option. Section 5 discusses the option that allows a mix of uncoordinated and uniform fire. Section 6 discusses the shoot-look-shoot attrition option, and Section 7 discusses the Lanchester squa-- attrition option.

3. Attrition Assuming Uncoordinated Fire

a. Assumptions

1) At a fixed time, all of the targets become vulnerable to attack by all of the shooters.

2) At this time, each shooter (i.e., each shooting weapon for each of its engagements) selects one target weapon (from among those present) to attack. Let the probability that a shooter of type i attacks a particular target weapon of type j be denoted by $\ddot{a}_{ij}(t_1,...,t_n)$ .en the target force consists of $t_{j'}$ weapons of type j', where

$$\sum_{j=1}^{n} \ddot{a}_{ij}(t_1,...,t_n)t_j = 1$$

for all i.

3) Given that a shooter of type i attacks a target of type j, the shooter kills that target (i.e., fires a lethal shot at the target) with probability p_{ij} for all i and j.

4) The target selection and firing processes of all of the shooters are mutually independent (so that, for example, two different shooters can choose to attack and can fire lethal shots at the same target--which results in one target being killed, not two).

Note that assumptions 2 and 4 imply that $a_{ij} = \ddot{a}_{ij} t_j$ for all i and j.

b. Resulting Attrition Equation

With these assumptions, the resulting number of targets of type j killed is a random variable. Let this random variable be denoted by $\Delta\tau_j$ and set

$$\Delta t_j = E[\Delta\tau_j]$$

for $j = 1,n$. If s_i and t_j are nonnegative integers for all i and j, then References [11], [17], and [18] show that these assumptions imply that

$$\Delta t_j = \begin{cases} t_j\left[1 - \prod_{i=1}^{m}\left(1 - \dfrac{a_{ij}p_{ij}}{t_j}\right)^{s_i}\right] & t_j > 0 \\ 0 & t_j = 0\,. \end{cases}$$

For specifics, see equation 3.11' of [11], equation 7 of [17], and equation 17' of [18], and set $d_i = 1$ in each of these equations. See also equation 22 of Reference [19].

Clearly Δt_j need not be an integer. Thus, following the structure described in Section A.2 above, the numbers of surviving weapons need not be integers after the first attrition assessment involving these weapons. (Even for the first attrition assessment, the numbers of shooters and targets involved need not be integers, depending on the input values for W, E, U, and C.) In order to prevent anomalies from occurring when the number of shooters or targets are not integral (such anomalies can occur if $0 < s_i < 1$ or $0 < t_j < 1$ for any relevant i or j), COMBAT calculates Δt_j using the formula

$$\Delta t_j = \begin{cases} t_j\left[1 - \prod_{i=1}^{m}(1 - p_{ij}\min(1, a_{ij}/t_j))^{\uparrow s_i}\right] & t_j > 0 \\ 0 & t_j = 0 \end{cases}$$

where $x^{\uparrow y}$ is defined as

$$x^{\uparrow y} = \begin{cases} x^y & y > 1 \\ 1 - y(1-x) & y \le 1 . \end{cases}$$

4. Attrition Assuming Uniformly Coordinated Fire

Two cases are considered below. The first case allows heterogeneous targets but assumes that the shooters are homogeneous ($m = 1$). Assumptions are stated for this case and then the resulting attrition equation (which can be derived from this assumption) is presented. The second case allows heterogeneous shooters as well as heterogeneous targets. The attrition equation presented for this second case is an heuristic extension of that for the first case, but is not rigorously derived from assumptions for this (heterogeneous shooter) case.

In the following, for any nonnegative number x, let $\lfloor x \rfloor$ denote the largest integer less than or equal to x (i.e., $\lfloor x \rfloor$ is the integer part of x), and let $<x> = x - \lfloor x \rfloor$ (i.e., $<x>$ is the fractional part of x).

a. Assumptions for Homogeneous Shooters and Heterogeneous Targets

1) There is one type of shooter ($m = 1$) but there can be multiple types of targets ($n \geq 1$). Since $m = 1$, let $s = s_1$, $a_j = a_j(t_1,...,t_n) = a_{1j}(t_1,...,t_n)$, and $p_j = p_{1j}$ for $j = 1,n$.

2) At a fixed time, all of the targets become vulnerable to attack by all of the shooters. At this time, each shooter (i.e., each shooting weapon for each of its engagements) selects one target to attack according to the following rules. (For simplicity, these rules assume that $t_j > 0$ for all j--cases where $t_j = 0$ for some j follow trivially.)

First, at least $\lfloor a_j s \rfloor$ shooters are assigned to attack targets of type j, which leaves

$$s - \sum_{j=1}^{n} \lfloor a_j s \rfloor$$

shooters yet to be assigned. Assign each of these additional shooters to target types in a random manner such that each target type is equally likely to be subject to one additional shooter and no target type is subject to two or more additional shooters--this is clearly possible since

$$0 \leq s - \sum_{j=1}^{n} \lfloor a_j s \rfloor < n .$$

Accordingly, the number of shooters that attack targets of type j is a random variable, say r_j, where

$$\text{Prob}\{r_j = x\} = \begin{cases} 1 - \langle a_j s \rangle & x = \lfloor a_j s \rfloor \\ \langle a_j s \rangle & x = \lfloor a_j s \rfloor + 1 \\ 0 & \text{otherwise} . \end{cases}$$

Second, given that x shooters are attacking targets of type j, assign at least $\lfloor x/t_j \rfloor$ shooters to attack each target of type j, which leaves

$$x - \lfloor x/t_j \rfloor t_j = \langle x/t_j \rangle t_j$$

shooters yet to be assigned. Assign each of these remaining shooters to particular targets of type j in a random manner such that each such target is equally likely to be subject to one additional shooter and no target is subject to two or more additional shooters--this is clearly possible since

$$0 \leq \langle x/t_j \rangle t_j < t_j .$$

Accordingly, the number of shooters that attack each target of type j is a random variable, say $\hat{r}_j$, where

$$\text{Prob}\{\hat{r}_j = y \mid r_j = x\} = \begin{cases} 1 - \langle x/t_j \rangle & y = \lfloor x/t_j \rfloor \\ \langle x/t_j \rangle & y = \lfloor x/t_j \rfloor + 1 \\ 0 & \text{otherwise} . \end{cases}$$

3) Given that a shooter attacks a target of type j, the shooter kills that target (i.e., fires a lethal shot at the target) with probability p_j.

4) The firing processes are independent of the target selection processes and are mutually independent of each other.

Note that the first part of assumption 2 is consistent with the definition of a_j in that the average fraction of engagements that shooters make against any target of type j is

$$\sum_{x=0}^{\infty} x\text{Prob}\{r_j = x\}/s =$$

$$(\lfloor a_j s \rfloor(1 - \langle a_j s \rangle) + (\lfloor a_j s \rfloor + 1)\langle a_j s \rangle)/s =$$

$$(\lfloor a_j s \rfloor + \langle a_j s \rangle)/s = a_j$$

as required by the definition of a_j. Note also that the second part of assumption 2 is consistent with uniform coordination of fire in that, if $a_j s$ is an integer, then

$$\lfloor a_j s/t \rfloor \le \hat{r}_j < \lfloor a_j s/t \rfloor + 1 .$$

b. Resulting Attrition Equation for Homogeneous Shooters and Heterogeneous Targets

With these assumptions, the resulting number of targets of type j killed is a random variable. Let this random variable be denoted by $\Delta\tau_j$ and set

$$\Delta t_j = E[\Delta\tau_j]$$

for j = 1,n. If the number of shooters, s, and the numbers of targets, t_j, are nonnegative integers, then it can be shown that the assumptions above imply that

$$\Delta t_j = t_j[1 - ((1 - p_j)^{\lfloor a_j s/t_j \rfloor})(1 - \langle a_j s/t_j \rangle p_j)] .$$

Since this equation does not exhibit anomalous behavior when s and t_j are not integers (but are still nonnegative), it can be used (without modification) to calculate attrition when, for the reasons stated above, s and t_j are not (necessarily) integers.

c. Assumptions for Heterogeneous Shooters and Targets

1) There can be multiple types of shooters and multiple types of targets.

2) At a fixed time, all of the targets become vulnerable to attack by all of the shooters. At this time, each shooter (i.e., each shooting weapon for each of its engagements) selects one target to attack in such a manner that no individual target is attacked by fewer than $\lfloor s_i a_{ij}/t_j \rfloor$ shooters of type i. Thus, this assumption accounts for

$$\sum_{j=1}^{n} \lfloor s_i a_{ij}/t_j \rfloor t_j$$

shooters of type i, but leaves the target selection process of the remaining

$$s_i - \sum_{j=1}^{n} \lfloor s_i a_{ij}/t_j \rfloor t_j$$

shooters of type i unspecified for all i.

3) Given that a shooter of type i attacks a target of type j, the shooter kills that target (i.e., fires a lethal shot at that target) with probability p_{ij} for all i and j.

4) The firing processes are independent of the target selection processes and are mutually independent of each other.

d. A Consistent Attrition Equation for Heterogeneous Shooters and Targets

Due to assumption 2, it is not possible to derive a specific attrition equation that satisfies the assumptions of Section c, above. Additional research is needed if it is desired to extend these assumptions in such a manner that is conceptually appropriate yet yields a computationally tractable attrition equation. (Such research has begun--see Reference [16] for details.) It is possible, however, to state attrition equations that are consistent with the assumptions of Section c; one such is as follows.

Let

$$h_j = \sum_{i=1}^{m} \langle s_i a_{ij}/t_j \rangle$$

and

$$\tilde{p}_j = \begin{cases} \sum_{i=1}^{m} \langle s_i a_{ij}/t_j \rangle p_{ij}/h_j & h_j > 0 \\ 0 & \text{otherwise}\,, \end{cases}$$

and set

$$\Delta t_j = t_j[1 - (\prod_{i=1}^{m} (1 - p_{ij})^{\lfloor s_i a_{ij}/t_j \rfloor} (1 - \tilde{p}_j)^{\lfloor h_j \rfloor} (1 - \langle h_j \rangle \tilde{p}_j))]\,.$$

In addition to being consistent with the assumptions stated in Section c above, this attrition equation has several other characteristics worth noting. First, it reduces to the rigorously derivable attrition equation of Section b where the shooters are homogeneous. Second, it is independent of the labeling of shooters and targets. For example, if all data associated with shooters of type i are interchanged with data for shooters of type i', then the numbers of targets killed of each type remain unchanged, and an analogous statement holds for interchanging data associated with targets of two different types. Third, it requires no additional inputs. Finally, it is computationally tractable.

5. Attrition Assuming a Mix of Uncoordinated and Uniformly Coordinated Fire

a. Assumptions

1) At a fixed time, all of the targets become vulnerable to attack by all of the shooters.

2) At this time, the fraction z_i of the shooters of type i each select one target to attack according to assumption 2 of Section 3.a above, and the remainder $(1-z_i)$ of these shooters each select one target to attack according to assumption 2 of Section 4.a above.

3) Given that a shooter of type i attacks a target of type j, the shooter kills that target (i.e., fires a lethal shot at the target) with probability p_{ij} for all i and j.

4) The independence assumptions of Sections 3.a and 4.c apply as appropriate, and the target selection and firing process of the uncoordinated shooters are mutually independent of the target selection and firing process of the uniformly coordinated shooters.

b. A Consistent Attrition Equation

In order to survive, a target must survive all of the $z_i s_i$ uncoordinated shooters of type i for all i, and also must survive all of the $(1-z_i)s_i$ uniformly coordinated shooters of type i for all i. Accordingly, it is reasonable and consistent to estimate the expected number of targets of type j that are killed, Δt_j, by

$$\Delta t_j = t_j(1 - \dot{q}_j \bar{q}_j)$$

where

$$\dot{q}_j = \prod_{i=1}^{m} \left(1 - \frac{a_{ij} p_{ij}}{\max\{1, t_j\}} \right) \uparrow z_i s_i$$

and

$$\tilde{q}_j = \prod_{i=1}^{m} (1-p_{ij})^{\lfloor (1-z_i)s_i a_{ij}/t_j \rfloor} (1-\bar{p}'_j)^{\lfloor h'_j \rfloor} (1-<h'_j>\bar{p}'_j)$$

for $j = 1,n$, where h'_j and $\bar{p}'_j$ are calculated as h_j and $\bar{p}_j$ were calculated in Section 3.d except that $(1-z_i)s_i$ is used in place of s_i wherever s_i appears in 3.d.

Note that Δt_j is not simply a convex combination of the expected attrition assuming uncoordinated fire and the expected attrition assuming uniformly coordinated fire.

6. Attrition Assuming a Shoot-Look-Shoot Firing Process

Two cases are considered below. The first case allows heterogeneous shooters but assumes that the targets are homogeneous ($n = 1$). Assumptions for this case are stated and then the resulting attrition structure is presented. The second case presents a simplistic extension of that attrition structure to handle cases in which both the shooters and the targets can be heterogeneous.

a. Assumptions for Heterogeneous Shooters and Homogeneous Targets

1) There can be multiple types of shooters ($m \geq 1$) but only one type of target ($n = 1$). Since $n = 1$, let $t = t_1$, $\Delta t = \Delta t_1$, $p_i = p_{i1}$, and note that the allocation of fire of shooters among target types plays no role (i.e., $a_{i1} = 1$ for all i). Assume that $s_1,...,s_m$, and t are all nonnegative integers.

2) At a fixed time, all of the targets become vulnerable to all of the shooters, but the shooters do not all fire at this time. Instead, the shooters attack one-at-a-time according to the following rules. Let

$$\bar{s} = \sum_{i=1}^{m} s_i ,$$

and label all of the shooters numerically, with labels running from 1 through $\bar{s}$, so that each shooter (i.e., each shooting weapon for each of its engagements) has its own numeric label. Let σ be a permutation of $\{1,...,\bar{s}\}$. That is,

$$\sigma(k) \in \{1,...,\bar{s}\} \text{ and } \sigma(k) \neq \sigma(k') \text{ if } k \neq k' ,$$

where k and k' range from 1 through $\bar{s}$. Shooter $\sigma(1)$ attacks first, followed by shooter $\sigma(2)$, and so on through shooter $\sigma(\bar{s})$. When it is a shooter's turn to attack, that shooter selects one target to fire upon from among the targets remaining alive when its turn comes. That is, each shooter knows the outcome of all previous engagements before it selects a target to attack, and it never attacks a target that was killed in a previous engagement. Since all targets are identical, the choice of target (from among those remaining alive) is irrelevant. If all of the targets are killed before all of the shooters have attacked a target, the remaining shooters do not fire.

3) Given that a shooter of type i ($i = 1,\ldots,m$) fires on a target, it kills that target with probability p_i.

4) The firing processes are independent of the target selection process and are mutually independent of each other.

b. Resulting Attrition Process for Heterogeneous Shooters and Homogeneous Targets

Given the assumptions above, it can be shown that the expected number of targets killed, Δt, is independent of the order of fire, σ. That is, if σ and σ' are two different permutations of $\{1,\ldots,\bar{s}\}$, then the assumptions above imply that the expected attrition given that the shooters fire in the order specified by σ equals the expected attrition given that the shooters fire in the order specified by σ'. (In general, the expected number of shooters that fire depends on the order of fire here, but not the expected attrition.)

If $t \geq \bar{s}$, then each shooter is guaranteed a (live) target, and so

$$\Delta t = \sum_{i=1}^{m} p_i s_i \ .$$

To calculate Δt when $t < \bar{s}$, consider the following structure. Since the expected attrition is independent of the order of fire, assume for simplicity that the shooters fire in order by type (with all type-1 shooters firing first, followed by all type-2 shooters and so forth). For $i = 1,\ldots,m$ and $\iota = 0,\ldots,t$, let $r_i(\iota)$ denote the probability that exactly ι targets remain alive after all of the shooters of type i have fired but (for $i = 1,\ldots,m-1$) before any shooters of type i+1 have fired, and set

$$r_0(\iota) = \begin{cases} 1 & \text{if } \iota = t \\ 0 & \text{otherwise .} \end{cases}$$

Then, starting with $i = 1$ and continuing through $i = m$, $r_i(\iota)$ can be calculated recursively using the formulas:

$$r_i(\iota) = \sum_{\iota'=\iota}^{t'_\iota} r_{i-1}(\iota')b(\iota'-\iota,s_i,p_i) \qquad \iota = 1,\ldots,t$$

and

$$r_i(0) = \sum_{\iota'=0}^{t'_\iota} r_{i-1}(\iota')\bar{b}(\iota',s_i,p_i)\ ,$$

where $t'_\iota = \min\{t,\ s_i + \iota\}$ for $\iota = 0,\ldots,t$,

$$b(\iota, s,p) = \binom{s}{\iota} p^s(1-p)^{1-s}$$

$$= \frac{s!p^s(1-p)^{1-s}}{\iota!(s-\iota)!}\ ,$$

and

$$\bar{b}(\iota', s,p) = \sum_{\iota=\iota'}^{s} b(\iota, s,p)\ .$$

Once values for $r_m(\iota)$ have been determined, Δt can be calculated by the formula:

$$\Delta t = \begin{cases} \sum_{i=1}^{m} p_i s_i & t \geq \bar{s} \\ t - \sum_{\iota=1}^{t} \iota r_m(\iota) & t < \bar{s}\ . \end{cases}$$

The relevant sections of the COMBAT computer program are based on these equations; however the code is somewhat more complex because it is also designed to handle cases in which the numbers of weapons involved are not (necessarily) integers.

c. A Simple Extension to Consider Heterogeneous Shooters and Targets

One way (perhaps the simplest way) to use a homogeneous attrition equation in a scenario that contains heterogeneous weapons is as follows. First, convert the heterogeneous weapons to an equivalent number of notional weapons of a single type by adding all the weapons (on the same side) together and by taking weighted averages of the effectiveness parameters. Second, use the homogeneous attrition equation with these homogeneous notional weapons. If the notional weapons in question are only shooting weapons (e.g., the target weapons are already homogeneous), then no third step is necessary. Otherwise (i.e., if this notionalization is being done either on both shooting and target weapons or on target weapons only), then a third step is needed. This third step is to prorate the notional losses among the various types of target weapons in some manner, e.g., in proportion to the capability of the shooting side to kill the various types of targets as given by

$$\hat{p}_j = \sum_{i=1}^{m} s_i a_{ij} p_{ij} .$$

An advantage of this approach is that it can always be applied, no matter how complex the homogeneous attrition equation is. A disadvantage of this approach is that, in a sense, it artificially converts a heterogeneous scenario into a homogeneous one and then applies a homogeneous attrition structure to that scenario, rather than extending the homogeneous structure to a structure that can directly address fully heterogeneous scenarios.

As currently coded, COMBAT uses a version of this simple approach to allow it to consider heterogeneous targets here. In particular, if n > 1, COMBAT computes a total number of notional target weapons, $\bar{t}$, as

$$\bar{t} = \sum_{j=1}^{n} t_j ,$$

and it computes a notional probability-of-kill by shooters of type i against those notional targets, $\bar{p}_i$, as

$$\bar{p}_i = \sum_{j=1}^{n} a_{ij} p_{ij}$$

for all i. These computations reduce the heterogeneous shooter and heterogeneous target case to a heterogeneous shooter and homogeneous target case, which allows the structure of Section b, above, to be applied. The resulting notional losses, say $\Delta\bar{t}$, are then prorated among target types in proportion to $\hat{p}_j$ as defined above, so that the number of targets of type j killed, Δt_j, is given by

$$\Delta t_j = \begin{cases} \left(\hat{p}_j / \sum_{j'=1}^{m} \hat{p}_{j'} \right) \Delta\bar{t} & \hat{p}_j > 0 \\ 0 & \text{otherwise} . \end{cases}$$

Note that this approach, while somewhat simplistic, is fully adequate to address the special case in which all probabilities of kill depend only on the type of shooter, not the type of target (i.e., $p_{ij} = p_i$ for all j), and all allocations of fire are in proportion to the numbers of targets present (i.e., $a_{ij} = t_j / \tilde{t}$ for all i).

d. Comments

First, concerning the heterogeneous shooter, homogeneous target case, the equations presented in Section b are not very computationally attractive. Perhaps more tractable formulas can be found. Second, while some effort was devoted to using a computationally efficient form of these equations in the COMBAT computer program, perhaps a much more efficient coding of these equations can be devised. Third, this portion of the code of COMBAT has not been extensively tested. Perhaps additional testing will uncover coding errors. Fourth, as discussed above, the extension to cover heterogeneous targets is quite simple, and it might not be adequate for some fully heterogeneous cases.

One way to address this last comment is to assume that shooters are preallocated to types of targets. That is, for each j, $s_i a_{ij}$ shooters of type i (for all i) would be allocated to attack targets of type j, and these shooters could not attack any other types of targets. This assumption is relatively target-favorable because, if a particular shooter is allocated to attack targets of type j and all of these targets are killed before that shooter's turn to fire arrives, then that shooter loses its chance to fire even if some other types of targets are still alive. This assumption is not currently available as an option in COMBAT, but it would be relatively easy to add and it also might improve the computational efficiency of this part of the code. Recent work on this approach for considering heterogeneous shoot-look-shoot fire is discussed in Reference [28].

7. Attrition Assuming a Lanchester Square Process

a. Assumptions

Various sets of assumptions can give rise to the Lanchester square attrition equation presented in Section b, below. All of these sets of assumptions are, in a reasonable sense, qualitatively different than the sets of assumptions given in Sections 3 through 6 above. Three such sets of assumptions are as follows.

First, attrition can simply be postulated to follow the Lanchester equation given below. (That is, it can just be assumed that attrition is adequately modeled by this equation.) Three of the flaws in making this (essentially tautological) assumption are: (1) it does not develop or derive the equation from relatively more fundamental assumptions concerning combat interactions, (2) it does not offer any insight as to the relative appropriateness either of the equation itself or of possible values for its effectiveness parameters, and (3) it inhibits comparisons with the assumptions that lead to the other attrition equations described above.

Second, the following set of assumptions could be made. This set of assumptions is divided into three subsets. The first subset consists of those assumptions that imply that attrition can be modeled as the heterogeneous Lanchester square versio.. of a continuous time discrete state space Markov process. See Section 2.4 of Reference [11] for a list of these assumptions. The second subset consists of the single assumption that the length of the time periods are sufficiently short and the numbers of weapons of all types on both sides are sufficiently large that the probability that any particular type of weapon is annihilated during any time period is negligible. This assumption, combined with theorem 2.8 of [11] (which is proved in Section 3 of [20]), means that attrition during a time period can be calculated by difference equations 2.15a and 2.15b of [11]. The third subset consists of the assumption that the length of the time period is sufficiently short that the attrition computed using these differential equations can be adequately approximated by attrition computed using corresponding difference equations.

The first subset of these assumptions are plausible, and can be compared and contrasted with the assumptions stated in Sections 3 through 6 above. The assumption that annihilation is negligible for each type of weapon might be reasonable for initial time periods, but grows less and less reasonable over time as simulated in a dynamic model of combat. As argued in Section A.4 above, the assumption that differential equations are adequately approximated by difference equations can be quite poor in practice (but would hold for models that are used to simulate attrition over very short time periods).

A third set of assumptions that yields the Lanchester square attrition equation given in Section b is to assume that a shoot-look-shoot structure applies, that shooters are preallocated to targets, and that the probability that any type of target is annihilated is negligible. Specifically, these assumptions are as follows.

1) At a fixed time, all targets of type j become vulnerable to s_1a_{1j} shooters of type 1 plus s_2a_{2j} shooters of type 2 plus...plus s_ma_{mj} shooters of type m, and these targets are only vulnerable to these shooters, where j = 1,...,n. Since these s_ia_{ij} shooters of type i attack only targets of type j and since the t_j targets of type j are attacked only by these shooters, this assumption implies that the overall attrition process being modeled consists of n separate attrition processes each of which can be heterogeneous in shooter types but is homogeneous in target type.

2) Assumption (2) of Section 6.a above applies to each of these n heterogeneous shooter, homogeneous target attrition processes. Thus, for each j from 1 through n, the targets of type j are vulnerable to a shoot-look-shoot attrition process in which the number of shooters of type i equals s_ia_{ij}.

3) Given that a shooter of type i fires on a target of type j, it kills that target with probability p_{ij}.

4) The firing processes are independent of the target selection processes and are mutually independent of each other.

5) For each j, either

$$t_j \geq \sum_{i=1}^{m} s_i a_{ij}$$

or the probability that all of the weapons of type j are killed is negligible.

Assumptions (1) through (4) are precisely the assumptions that would be made for the preallocated shoot-look-shoot process described in Section 6.c above. The problems here concern assumption (5). First, note that assumptions (1) through (4) completely define the attrition process and so, given these assumptions and the relevant data, the probability that weapons of type j are annihilated is fixed. Accordingly, if the phrase "is negligible" in assumption (5) were made specific (e.g., is less than 0.01), then assumption (5) would either be true or be false. That is, it would either follow for assumptions (1) through (4) and the data, or it would contradict these assumptions and data. Either way, it would not be a separate assumption. (This same comment also applies to the negligibility-of-annihilation assumption made concerning the continuous time attrition assumptions discussed earlier in this section.) Second, as in the discussion above, the probability that any given type of target weapon is annihilated might be negligible for initial time periods, but this probability will (in general) grow over time, and so (while perhaps initially reasonable) assumption (5) grows less and less reasonable over time as simulated in a dynamic model of combat. Third, note that the unilateral attrition equation discussed here will be applied to side 1 shooting at side 2 and to side 2 shooting at side 1. Thus, if the numbers of weapons of all types on side 2 (for example) are quite large relative to the total number of weapons on side 1, then when side 1 shoots at side 2 the probability that any given type of weapon on side 2 is annihilated would be quite small (perhaps zero). However, in this example, when side 2 shoots at side 1, many weapons will be shooting at relatively few weapons, and so the probability that some type of weapon on side 1 is annihilated might be significant. That is, using the two-sided notation of Chapter II, the more likely it is that assumption (5) is satisfied for side s shooting at side s', the less likely it may be that this assumption is satisfied when side s' shoots at side s (for either value of s). Finally, note that if the time period over which attrition is assessed can be adjusted, then assumption (5) can always be satisfied by sufficiently reducing the length of this time period. In particular, reducing the length of this time period lowers the engagement rates of the shooters, which lowers s_i in (5) without lowering t_j. However, depending on the model and data being used, it .ay not be computationally practical to make separate attrition assessments for each time period when that time period is very short relative to the length of the combat being simulated.

b. Resulting Attrition Equation

If the probability of annihilation of any type of target is negligible, then the assumptions discussed above imply that (except for negligible cases) when it is any shooter's turn to attack a target, there is always at least one target (of the appropriate type) still alive for it to attack. Accordingly, if the negligibility-of-annihilation assumption holds, then the other assumptions (of the second or third set of assumptions) listed above imply that the number of targets of type j killed is approximately equal to Δt_j where

$$\Delta t_j = \sum_{i=1}^{m} s_i a_{ij} p_{ij} .$$

Since the negligibility-of-annihilation assumption may not hold, COMBAT does not use this equation directly. Instead, if the Lanchester square attrition option is selected, COMBAT computes Δt_j using the equation

$$\Delta t_j = \min\{t_j, \sum_{i=1}^{m} s_i a_{ij} p_{ij}\}$$

for all j.

In Section 6.d above, it was argued that preallocating shooter types to target types was relatively target favorable because shooters might lose their chance to fire due to annihilation of the type of target they were allocated against while other types of targets still remain alive. Part of the basis for this argument is that probabilities of annihilation are being addressed correctly. This is clearly not the case here, and so this argument does not apply. Indeed, the opposite applies. That is, the Lanchester square attrition equation stated just above is shooter favorable in that each shooter is assumed to be able to shoot at a live target (of the appropriate type) even if, due to previous shooters, no targets (of that type) would be alive when that shooter's turn to fire arrived if probabilities of annihilation were being considered correctly. This is the basis for the argument made in Section 2 above that Lanchester square attrition can be viewed as representing an (impossible to actually achieve) upper bound on the attrition that would result from the highest level of coordination possible (i.e., shoot-look-shoot) among the shooters.

8. Summary

The attrition equations presented above differ primarily in their assumptions concerning the degree of coordination among shooters. Attrition equations are presented for: (a) an uncoordinated firing process, (b) a process that is a mixture of uncoordinated and uniformly coordinated fires, (c) a uniformly coordinated firing process, (d) a perfect shoot-look-shoot firing process, and (e) a Lanchester square firing process (which can be viewed as giving a computationally tractable upper bound on the attrition that would result from a shoot-look-shoot firing process).

a. Relationships Among These Attrition Equations

As the degree of coordination among the shooters increases, the number of kills they achieve increases. In particular, let

Δt_j^n = Δt_j assuming uncoordinated fires (as in Section 3 above),

Δt_j^m = Δt_j assuming a mixture of uncoordinated and uniformly coordinated fires (as in Section 5 above),

Δt_j^c = Δt_j assuming uniformly coordinated (i.e., even) fires (as in Section 4 above),

Δt_j^p = Δt_j assuming a perfect shoot-look-shoot firing process (as in Section 6 above), and

Δt_j^l = Δt_j assuming a Lanchester square firing process (as in Section 7 above).

Then it follows that

$$\Delta t_j^n \le \Delta t_j^m \le \Delta t_j^e \begin{cases} \le_? \Delta t_j^p \le_? \Delta t_j^l \\ \le \Delta t_j^l , \end{cases}$$

for j = 1,...,n, where the questionable inequalities (denoted by "$\le_?$") hold for heterogeneous shooters and homogeneous targets (n = 1), but may not hold for heterogeneous targets due to the simplistic averaging structure used in Section 6.c above.

As indicated in the discussions above, one way to more carefully address heterogeneous targets in a shoot-look-shoot firing process would be to preallocate the shooters to types of targets and then allow shooters to engage the type of target they are allocated against using a shoot-look-shoot process. As an option, shooters could also be preallocated against types of targets in the uncoordinated firing process. (In essence, shooters are already preallocated against types of targets in both the uniformly coordinated firing process and the Lanchester square firing process, so additional options concerning preallocation are not possible for these processes.) If the code of COMBAT were changed to allow full preallocation of shooters to types of targets, then COMBAT would also produce

$\Delta t_j^{\bar{n}}$ = Δt_j assuming preallocated fires against target types, where the fires against each type of target are uncoordinated,

$\Delta t_j^{\bar{m}}$ = Δt_j assuming preallocated fires against target types, where the fires against each type of target are a mixture of uncoordinated fires and uniformly coordinated fire, and

$\Delta t_j^{\bar{p}}$ = Δt_j assuming preallocated fires against target types, where the fire against each type of target follows a perfect shoot-look-shoot firing process.

Then it is conjectured that

$$\Delta t_j^{n} \le \Delta t_j^{\bar{n}} \le \Delta t_j^{\bar{m}} \le \Delta t_j^{e} \le \Delta t_j^{\bar{p}} \begin{cases} \le_? \Delta t_j^{p} \le_? \Delta t_j^{l} \\ \le \Delta t_j^{l} , \end{cases}$$

for j = 1,...;n, where $\le_?$ is used as described above.

A useful aspect of these inequalities is as follows. If, in a particular analysis,

$$\Delta t_j^{n} \cong \Delta t_j^{l}$$

for all relevant j, then the degree of coordination (as described here) among the shooters does not play a significant role concerning attrition in that analysis. Also, if

$$\Delta t_j^{e} \cong \Delta t_j^{l}$$

for all relevant j, then it would likely be a waste of effort to perform the potentially extensive calculations required to compute attrition using the formulas for perfect shoot-look-shoot fire. Conversely, if, in an analysis,

$$\Delta t_j^{n} << t_j^{l}$$

for some j, then assumptions concerning the degree of coordination among the shooters could be quite important in that analysis.

b. Other Sets of Assumptions

As stated at the beginning of this section, the sets of assumptions made here differ primarily in how they treat coordination among shooters. Other sets of assumptions that differ in other ways could also be investigated and added to COMBAT (or used in other dynamic models of combat) if desired.

For example, sets of assumptions to explicitly model area fire could be postulated, and attrition equations that are based on these area-fire assumptions could be used. (See, for example, Reference [16].) Alternatively, sets of assumptions to explicitly model barrier penetration processes could be postulated, and attrition equations based on these barrier-penetration assumptions could be used in models less aggregated than COMBAT. (See, for example, References [14], [21], and [22].)

In short, Sections 3 through 7 above discuss only a few of the possible sets of assumptions that might yield attrition equations which could be used in deterministic estimation models as described in Section A above.

C. CONVERTING UNILATERAL ATTRITION ASSESSMENTS INTO BILATERAL ATTRITION

Each of the attrition equations presented in Section B above is defined in terms of "shooters" on one side versus "targets" on the other. Two-sided models need to use attrition equations that consider weapons that can simultaneously be both shooters (killing weapons on the other side) and targets (being killed by those enemy weapons). Simulating weapons (on each side) that are both lethal and vulnerable, instead of invulnerable shooters on one side versus impotent targets on the other, was accomplished in older models in the following manner: First, the initial numbers of weapons on one side were used as "shooters" in a unilateral attrition equation (such as one of those presented in Section B) to calculate the numbers of weapons killed on the other side. Then, before these kills were assessed, the initial numbers of weapons on the other side were used as "shooters" in a unilateral attrition equation to calculate the numbers of weapons killed on the first side. After both of these calculations, all kills were assessed. This procedure has been pejoratively described as modeling "all bullets passing in mid-air." If attrition were assessed only over very short time periods (so that there would be very few engagements per time period), then this procedure would not be unreasonable. However, as argued above, it is not usually practical to assess attrition this frequently. Conversely, this procedure is generally unreasonable (and can significantly overstate numbers of kills) when used with time periods that, while computationally tractable, are sufficiently long that many engagements can occur within any one period.

This old procedure can be reproduced as an optional special case of the more general procedure discussed here. However, the procedure presented here also allows options that avoid this "bullets passing in mid-air" characteristic. An outline of this more general procedure is as follows.

Unilateral attrition equations are used four times: first for the initial side 1 weapons shooting at side 2, second for the surviving (from that first assessment) side 2 weapons shooting back at side 1, third for the initial side 2 weapons shooting at side 1, and fourth for the surviving side 1 weapons shooting back at side 2. The overall attrition (for each side) is then computed as averages of the attrition from the "side 1 shoots first" case and the "side 2 shoots first" case. Of course, in real battles, it is unlikely that either side would fire all of its "shots" before the other side shoots even once; the averaging approach used here is intended to provide a relatively reasonable and tractable method for representing the average results of individual engagements.

Section 1 below presents the notation and specific formulas for this procedure, and Section 2 discusses some of its characteristics.

1. Notation and Equations

The unilateral attrition equations of Section B above can be viewed as computing (for each j) the number of targets killed of type j as a function of the number of shooters and targets of all types. Accordingly, these unilateral attrition equations can be written in a generic form (using one-sided notation) as

$$\Delta t_j = f_j(s_1,\ldots,s_m;t_1,\ldots,t_n) \qquad j = 1,n \ ,$$

or

$$\Delta t_j = f_j(s;t) \qquad j = 1,n \ .$$

Using two-sided notation, this generic form can be written as

$$C_j^{s'} = f_j^{s'}(W_1^s,\ldots,W_{N^s}^s;W_1^{s'},\ldots,W_{N^{s'}}^{s'}) \qquad j = 1,N^{s'} \ ,$$

or

$$C_j^{s'} = f_j^{s'}(W^s;W^{s'}) \qquad j = 1,N^{s'} \ ,$$

where $C_j^{s'}$ is the number of weapons lost of type j on side s' as computed by a unilateral attrition assessment, the shooting side has W_i^s weapons of type i (i = $1,N^s$) available to make engagements, there are $W_{j'}^{s'}$ vulnerable weapons of type j' (j' = $1,N^{s'}$), and (as always for two-sided notation) s = 1,2 and s' = 3–s.

For s = 1,2 and s'' = 1,2, let

$$B_j^{s'}(s'') = \begin{cases} f_j^{s'}(W^s;W^{s'}) & s'' = s \\ \\ f_j^{s'}(W^s - B^s(s');W^{s'}) & s'' = s' \ , \end{cases}$$

where j = $1,N^{s'}$ and s' = 3–s. That is, $B_j^{s'}(s'')$ is the number of weapons of type j on side s' that are lost when side s'' shoots first. Then COMBAT computes the overall number of weapons of type j on side s' that are lost, $D_j^{s'}$, by the formula

$$D_j^{s'} = \begin{cases} yB_j^{s'}(1) + (1-y)B_j^{s'}(2) & 0 \le y \le 1 \\ \left(\frac{4-y}{2}\right)B_j^{s'}(s') + \left(\frac{y-2}{2}\right)B_j^{s'}(s) & 2 \le y \le 4\,, \end{cases}$$

where y is essentially an input to COMBAT such that $y \in [0,1] \cup [2,4]$. (Strictly speaking, y corresponds to the working variable Y in COMBAT, and this working variable is calculated by

$$Y = \begin{cases} \text{ZROBSF} & 0 \le \text{ZROBSF} \le 1 \text{ or } 2 \le \text{ZROBSF} \le 4 \\ 0.5 & \text{otherwise}\,, \end{cases}$$

where ZROBSF is an input to COMBAT.)

2. Discussion

The procedure presented in Section 1 above is based on ideas developed in Reference [23] (see especially Sections B.3.b and B.3.c of [23]) and in Reference [24]. The interested reader should consult these references for details, theory, and examples.

Note that setting y = 0.5 (or y = 3) results in computing $D_j^{s'}$ (for both s') as the arithmetic mean of $B_j^{s'}(1)$ and $B_j^{s'}(2)$. See [23] and [24] for rationale for this value of y.

If $0 \le y \le 1$ then y represents the fraction of engagements in which side 1 shoots first. Setting y = 1 is quite side 1 favorable in that $D_j^{s'}$ is set equal to $B_j^{s'}(1)$ for both s', while setting y = 0 is quite side 2 favorable in that $D_j^{s'}$ is set equal to $B_j^{s'}(2)$ for both s'.

If $2 \le y \le 4$ then y represents the degree to which potential kills suppress lethal fire. Setting y = 2 means full suppression (i.e., setting y = 2 gives one way of incorporating a "fear of death" into the model). Setting y = 4 means no suppression (i.e., setting y = 4 reproduces the "all bullets pass in mid-air" procedure of older models).

It should be noted that a somewhat more general version of this procedure is suggested in Reference [25] and has been incorporated into the model described in Reference [26]. This yet-more-general procedure also allows weighted averages of $B_j^{s'}(1)$ and $B_j^{s'}(2)$ to be used to calculate $D_j^{s'}$, where the weighted averages are determined by ratios involving the weighted numbers of weapons present on each side--see [25] or Appendix A of [26] for details.

Further extensions of this procedure are also possible. For example, as suggested in [23], some of the weapons on the side shooting first could be allowed to (explicitly) suppress but not kill enemy weapons, so that such suppressed enemy weapons could not fire back that time period. Such an extension would require new inputs, but would be easy to incorporate into the "shoot-then-shoot-back" structure used in COMBAT and in [26].

VI. ON USING COMBAT AS A DYNAMIC SIMULATION OF WARFARE

As discussed in Chapter I, COMBAT has several potential uses, one of which is as a highly aggregated, general purpose model of conventional combat. However, since COMBAT was not primarily designed for this purpose, it omits many features that might be desirable for such a model. This chapter discusses the generally more important of these omitted features.

Section D of Chapter II discusses two ways that combat aircraft can be addressed in COMBAT. One of these, treating ground attack aircraft as non-vulnerable weapons systems, applies only to one approach for calculating scores for these systems--it does not apply to the calculation of attrition or to the dynamic simulation of warfare. When COMBAT is being used as a dynamic model of combat, aircraft must be considered as described in Section D.1 of Chapter II. That is, if combat aircraft are to be addressed in COMBAT, they must be considered as comprising one or more types of weapons systems contained in the N^s types of fully interacting systems being considered for side s. Accordingly, throughout this chapter, aircraft will be considered in this manner.

Section 1 below discusses some characteristics and limitations that apply to both ground and air combat; Section 2 discusses those that essentially apply to air combat only.

A. LIMITATIONS THAT AFFECT BOTH GROUND AND AIR COMBAT

1. Resources

a. Replacements

COMBAT does not simulate war reserve stocks from which weapons systems could be drawn to replace weapons that suffer attrition in combat. Such replacement stockpiles could be added in a relatively straightforward manner, if desired. Modeling issues concerning such an addition include: (1) determining an upper bound on the number of replacement weapons systems that units in combat or wings on operating air bases could accept (assuming there are sufficiently many such weapons systems in replacement stockpiles), and (2) determining how quickly such replacement weapons systems could enter combat.

b. Munitions

COMBAT does not account for the consumption of munitions or the impact of shortfalls of selected types of munitions. Accounting for munitions properly requires, at a minimum, that: (1) probabilities-of-kill are functions of the types of munitions being used, (2) when munitions of one type run out, then other types (if available) are used instead, and (3) when munitions of all of the types that a weapons system can use run out, then that weapon has no effectiveness. Significant effort would be required to properly add these considerations to COMBAT--it might be more efficient to add relevant features of COMBAT to other models that already address munitions. Nevertheless, the proper consideration of munitions is, in many cases, quite important. Recent work on the explicit consideration of multiple types of munitions in attrition equations is discussed in Reference [16].

c. People and Supplies

COMBAT does not account for personnel or supplies. Therefore it does not account for the wounding or killing of people, or for the consumption or destruction of supplies, or for the impact of shortfalls of personnel or supplies on the effectiveness of weapons systems.

2. Interactions

a. Battle Damage and Repair

COMBAT does not account for the possibility that weapons systems are damaged but not destroyed by enemy fire. Accounting for damaged systems in a reasonable manner implies that the repair and return to combat of such systems should also be simulated in some suitable manner.

b. Non-Battle Damage and Destruction

COMBAT does not account for non-battle losses. Accounting for non-battle damage in a reasonable manner requires simulating the repair of weapons systems as described above. Accounting for non-battle destruction would be easy to add, if desired.

c. Effectiveness Parameters that are Functions of Attack and Defense

Many models of combat allow some of their input effectiveness parameters to depend on whether side 1 is attacking side 2 or vice versa. COMBAT does not contain a method for determining the frequency that each side is on the attack, and none of its input effectiveness parameters are functions of whether either side is on attack or defense. These concepts could be added to COMBAT if desired. An outline of a reasonable way to make such an addition is given in Appendix B.

d. Degrading the Nominal Effectiveness of Unbalanced Forces

It can be argued that the nominal effectiveness of a force should be degraded if that force is significantly unbalanced. For example, a force consisting only of artillery and aircraft could be considered as being unbalanced and hence as not being an effective force because it could easily be overrun by enemy armor and infantry. Conversely, a more balanced force consisting of fewer artillery and aircraft (but with some armor and infantry) could be quite effective because the armor and infantry, in addition to providing their own firepower, help protect the artillery and air bases. This type of degradation of effectiveness is not currently simulated in COMBAT. A reasonable way to roughly represent this type of loss of effectiveness is described in Appendix C.

e. Area Fire

As discussed in Chapter V, COMBAT does not explicitly simulate area fire. Reference [16] gives a reasonably general structure for modeling area fire that is methodologically consistent with the aimed fire structure presented above.

3. Geography

a. Geography and Attrition

COMBAT does not simulate the impact of variations in terrain, prepared defenses, or any other aspect of what might loosely be called geography in its attrition calculations.

b. Depth from the Front

COMBAT does not explicitly locate weapons systems as being at various depths from the front. Thus, for example, no transportation capability is required to move weapons from rear areas to active combat. The fact that all weapons are not always in contact with the enemy can be represented as follows. Let

α_i^s = the average fraction of weapons of type i on side s that are in contact with the enemy, where i = 1,N^s and s = 1,2.

Then replacing E_i^s (as defined in Section B of Chapter II) with $\alpha_i^s E_i^s$ and replacing U_i^s (as defined in Section B.1 of Chapter V) with $\alpha_i^s U_i^s$ reduces both the lethality and the vulnerability of weapons systems to account for those that, on average, are not in contact with the enemy.

c. Maneuver

'?OMBAT does not simulate maneuver directly nor does it simulate the impact of maneuver on attrition. It would (in general) be easier to add selected features of COMBAT to a suitable model that simulates maneuver than it would be to add the simulation of maneuver to COMBAT.

d. Sectors

COMBAT allows the combat arena to be subdivided into an input number of identical combat areas (or sectors), where this input is denoted by C in Section B.1 of Chapter V above. A reasonable extension of this structure is to (optionally) subdivide the combat arena into non-identical combat areas such that the input fraction F_{ic}^s of the weapons systems of type i on side s would be located (and fight) in combat area c, where

$$0 \leq F_{ic}^s \leq 1$$

for c = 1,...C, and

$$\sum_{c=1}^{C} F_{ic}^s = 1$$

for i = 1,...,N^s and s = 1,2.

Note, however, that it may not be appropriate to attempt to include more detailed assignments than this here. For example, two attempts to consider such additional detail that are contained in some other large-scale ground models are as follows. (These considerations are not contained in COMBAT or in the model of Reference [26].)

First, some other models account for ground weapons as being integral parts of units in that all ground weapons are located wherever their unit is located and the only way to move these weapons is to move their units (which necessarily moves all of the weapons in those units). In reality, of course, not all weapons in a large unit are located near where that unit's headquarters is located; brigades can move separately from the division they belong to, battalions can move separately from the brigade they belong to, and so forth. More importantly, grouping weapons into large units creates decision-making problems for fully automated models that can lead to logical anomalies. For example, suppose that force A differs from force B only in that A has some additional combat units in it that B does not have. Suppose that a particular model moves units into or out of combat in one way when evaluating A but moves these units in a different way when evaluating B. Then it can turn out that the model in question produces results for B that are better (for the side in question) than its results for A due to these different movement decisions, even though force B is strictly inferior to force A.

Second, some models divide the theater (or region of interest) laterally into parallel sectors. Sometimes these sectors are hidden under an overlay of hexagons; sometimes these sectors are clearly displayed with the forces on either side moving in each of them like a piston. While this structure in itself causes no problems, problems do occur when this structure is combined with the characteristic just described concerning grouping ground weapons into units. In particular, this structure exacerbates the decision-making problems discussed above because now, not only do units have to be moved into and out of combat, they have to be moved to and from particular sectors, and decision rules must (at least implicitly) exist to move units from sector to sector. If it were possible to determine decisions that were optimal from a game-theoretic viewpoint, then a sector-type model with these optimal decision rules could be used to measure the effectiveness of alternative forces. However, since such optimal rules cannot generally be found, these models use (frequently complex) heuristic rules to move forces, and the impact of those rules can dominate the impact of the capabilities of the forces that these models are attempting to assess.

For example, suppose that such a model is being used to assess two forces, A and B. Suppose that the results of running the model with A are significantly but not overwhelmingly better for the side in question than the results are with B. Then A might be overwhelmingly better than B but the decisions made for A were not as good as those made for B; or B might actually be better than A but the decisions made for A were so much better than those made for B that A appeared to be better than B according to the results of the model (these results being necessarily dependent both on the quality of the forces involved and on the decisions made for those forces). In the structure suggested above (using the proposed new input F^{s}_{ic}), this would be much less likely to happen (and in many reasonable cases, perhaps even all reasonable cases, it cannot happen) because this structure locates and moves (fractional) weapons, not units, among the various combat areas.

4. Output Measures

a. Killer-Victim Scoreboards

The algorithm used by COMBAT to compute killer-victim scoreboards is rather elementary. While this algorithm may be sufficient for many purposes, more complex (and contextually specific) algorithms have been proposed. Additional research to investigate the properties of these algorithms could be performed. See Section B.4.g of Chapter III of Reference [26] and (all of) Reference [29] for details.

b. FEBA Movement

COMBAT does not simulate the capture or loss of territory. That is, it does not simulate the movement of the Forward Edge of the Battle Area (FEBA) or (synonymously here) of the Forward Line of Own Troops (FLOT). This could be done, for example, as described in Section B.4.f of Chapter III of [26] (see also Section C.4 of Chapter VI of that reference). If the simulation of FEBA movement is added to COMBAT, it might also be desirable to incorporate and relate bounds on the attrition being suffered to the average rate of this movement. A discussion of a general method for doing this is given in Appendix D.

c. Different Uses for Different Scores

COMBAT can use weapons scores in essentially two ways. First, it uses these scores to calculate force ratios and other output measures. Second, it can (optionally) use such scores to calculate allocations of fire. There is no immutable reason why the scores used to compute output measures need to be the same (or even be computed by the same method) as the scores used to compute allocations of fire. For example, the scores used to compute allocations of fire could vary over time as discussed in Chapter IV above, while the scores used to compute output measures could be computed once and then held fixed throughout an analysis. Currently, COMBAT does not allow two different sets of weapons scores to be used in these two different ways within the same run of the model.

5. Changing Data Values During a Simulation

As discussed in Section A of Chapter V, when COMBAT is being used as a dynamic simulation of warfare, it simulates the passing of time by stepping through time in intervals of fixed length. These intervals are frequently called time periods, and many other models also process time in this manner. In all such models, it is generally desirable to be able to change the value of any input at the beginning (or, essentially equivalently, at the end) of any time period.

For example, some inputs concern the numbers of weapons systems initially in the combat arena, and the model changes the values of these inputs over time to account for attrition. For such inputs, it is generally desirable to be able to increment their current values at any time period to account for new weapons that enter the combat arena at that time period (and to decrement their values to account for non-simulated losses or transfers out of the combat arena). Other inputs concern the effectiveness of weapons systems, and usually models do not automatically change the values of these inputs over time. For such inputs, it is generally desirable to be able to replace their current value with a new value at any time period for any of a number of reasons.

COMBAT is not currently capable of accepting any change to any of its input values during the course of a run of the model.

6. User Interface

The input structure and output formats of COMBAT are quite rudimentary. While adequate for research purposes involving relatively small numbers of weapon types, these structures and formats would not be suitable for analyses involving large numbers of different types of weapons.

B. LIMITATIONS THAT PRIMARILY AFFECT AIR COMBAT

The major limitations of using COMBAT (as currently coded) to dynamically simulate air warfare can be grouped into four categories. First, COMBAT requires a fixed assignment of aircraft to missions. Second, non-lethal suppression cannot be modeled. Third, details concerning the timing of various air interactions within a time period are not representable. Finally, some potentially important characteristics of deep strikes are not representable. Each of these is discussed, in turn, below.

1. Fixed Assignments of Aircraft to Missions

There are many different missions to which combat aircraft can be assigned. For example, Reference [26] considers the following 12 types of combat missions:

MISSIONS FOR AIRCRAFT

Offensive Missions

Close air support--direct attack
Close air support--escort
Close air support--SAM-suppression
Interdiction--direct attack
Interdiction--escort
Interdiction--SAM-suppression
Air base attack--direct attack
Air base attack--escort
Air base attack--SAM-suppression
Belt SAM-suppression

Defensive Missions

Battlefield defense (i.e., area defense in front of the SAM belt)
Air base defense (i.e., area defense behind the SAM belt)

In reality, any combat aircraft could be assigned to fly one of at least two of these missions; many types of combat aircraft could fly several types of these missions, and some types of combat aircraft could fly all of these types of missions. Further, aircraft can (in general) change missions on any day of the war. As described in Section D.1 of Chapter II, COMBAT requires that each aircraft be assigned to a particular mission and no mission assignments can be changed over the course of the war being simulated.

2. Non-Lethal Suppression

COMBAT cannot currently simulate non-lethal suppression of any kind. Systems either are fully operational and continue to perform their mission (assuming input average effectiveness parameters) or have been destroyed by enemy fire. For example, in COMBAT, SAMs cannot be suppressed without also being killed by enemy SAM suppression aircraft, aircraft cannot be forced to jettison their ordnance and return home alive (but unsuccessful) due to enemy defenses, and runways on air bases cannot be attacked thereby (potentially) suppressing the aircraft on those bases (until the runways are repaired) without killing those aircraft. In reality all of these types of suppression are possible.

3. Timing of Interactions

In reality, the order in which air interactions occur during any period of time can be important. For example, escort aircraft want to engage enemy defense aircraft before those defenders can engage the attacking aircraft being escorted. Defense aircraft want to engage enemy attacking aircraft before those attack aircraft can deliver their ordnance against ground targets. Similarly, surface-to-air missile (SAM) suppression aircraft want to engage enemy SAMs before those SAMs can engage (direct) attack aircraft, and SAMs want to engage enemy attack aircraft before those attack aircraft can deliver their ordnance against ground targets. Also, geography dictates certain orderings. For example, aircraft on air base attack missions would (in general) first be vulnerable to enemy battlefield defense aircraft and SAMs in the combat area, then be vulnerable to enemy SAMs in the SAM belt, then be vulnerable to enemy air base defense aircraft, then be vulnerable to point-defense SAMs at the enemy air bases they are attacking.

As currently coded, COMBAT essentially assumes that all interactions occurring during a time period are evenly spread over the duration of that time period. That is, COMBAT can be used to simulate all of the interactions described just above, but not the order in which these interactions are likely to occur during a time period.

4. Selected Characteristics of Deep Strikes

a. Interdiction Missions

As currently coded, COMBAT cannot meaningfully simulate interdiction missions.

b. Air Base Attack Missions

In reality, if a side has fewer shelters than aircraft, then some aircraft would (in general) be unsheltered when their air base is attacked and hence be more vulnerable to those attacks than sheltered aircraft. Conversely if, over the course of combat, aircraft have suffered sufficient attrition (in the air) that a side eventually has more shelters than aircraft, then the extra (empty) shelters could serve as decoys (as in a "shell game") to reduce the effectiveness of an enemy air base attack. As currently coded, COMBAT essentially must assume that a fixed percentage of each type of aircraft is sheltered throughout the war being simulated. In addition, as noted in Section 2 above, COMBAT cannot meaningfully simulate attacks on runways.

5. Summary

With sufficient ingenuity, COMBAT can be used to simulate several aspects of air warfare in a highly aggregated manner. Significant effort would be required to allow COMBAT to simulate either additional aspects or more detail concerning air combat.

VII. SUMMARY DOCUMENTATION OF THE COMBAT COMPUTER PROGRAM

The purpose of this chapter is to document the COMBAT computer program sufficiently well so that an interested reader can relate the notation and methodological descriptions given above to the computer variables and procedures used in the code, and can meaningfully run this program.

A. INPUTS

1. Input Data

All of the inputs to COMBAT are listed (in alphabetical order) on Table VII-1. Most of these inputs have been defined and discussed in the chapters above. For each such input, Table VII-1 gives the corresponding algebraic notation (if any) that is used for that input above, and it gives the primary location (by chapter and section) where that input is defined and discussed. That chapter and section should be consulted for the definition of that input.

Table VII-1 also gives the conditions under which values for each of the inputs are required--values for some inputs are required only for certain values of other inputs. If values for an input are not required, these values must not appear in the input data file.

The inputs to COMBAT that were not defined in the chapters above are defined on Table VII-2. (Dashes in the "Notation" and "Chapter & Section" columns on Table VII-1 indicate that the input in question was not defined and discussed above.) In addition, definitions for a few inputs that were discussed above but are closely related to these newly defined inputs are also given on Table VII-2. All of the inputs that appear on both Table VII-1 and Table VII-2 are marked with an asterisk on Table VII-1.

Table VII-3 repeats the definitions given on Table VII-2 for "multiple choice" inputs that concern weapons scores and allocations of fire. Table VII-3 is included so that the definitions of these interrelated inputs can be listed in one place.

Table VII-1. Relationship Between the FORTRAN Inputs to COMBAT and the Algebraic Notation of Chapters II through V

FORTRAN Input	Algebraic Notation	Chapter & Section Containing Definition	Conditions Under Which Input Is Required
AC(I,IS)	$\underline{W}_i^s$	II.D.2.a	NA(IS) ≥ 1
CAREA	C	V.B.1	always
E(I,IS)	E_i^s	II.B	always
EA(I,IS)	$\underline{E}_i^s$	II.D.2.a	NA(IS) ≥ 1
FILEIN*	--	--	always
IALMTH(IS)*	a	III.A.2	always
IALVAL(IS)*	none	IV.A	always
IALWT*	--	--	always
IALWTA*	--	--	NA(•) ≥ 1
IATRTE(IS)*	none	V.B.2	always
IFVUL*	--	--	always
INVAL*	v	II.B	always
IRSCRN*	--	--	always
IVSUMP*	--	--	INVAL = 6
IWSCRN*	--	--	always
IWSCRO*	--	--	IRSCRN = 1 and IWSCRN = 2
I1	h	II.B	always
N(IS)	N^s	II.B	always

(continued)

Table VII-1. (Continued)

FORTRAN Input	Algebraic Notation	Chapter & Section Containing Definition	Conditions Under Which Input Is Required
NA(IS)	$\underline{N}^s$	II.D.2.a	always
NAMVAR*	--	--	always
NPER*	T	II.E.1	always
NWNAME*	--	--	IRSCRN = 1
P(I,J,IS)	P_{ij}^s	II.B	always
PA(I,J,IS)	$\underline{P}_{ij}^s$	II.D.2.a	NA(IS) ≥ 1
Q(I,IS)	q_i^s	III.D	IALMTH(IS) = 2 or 3
QA(I,IS)	$\underline{q}_i^s$	III.D	NA(IS) ≥ 1 and IALMTH(IS) = 2 or 3
TPCLAA(I,J,IS)	$\underline{\hat{A}}_{ij}^s$	III.B	NA(IS) ≥ 1 and IALWTA = 1
TYPCLA(I,J,IS)	$\hat{A}_{ij}^s$	III.B	IALWT = 1
TYPCLW(I,IS)	$\hat{W}_i^s$	III.B	IALWT = 1
U(I,IS)	U_i^s	V.B.1	IFVUL = 1
V(I,IS)	none	II.F	INVAL = 0
W(I,IS)	W_i^s	II.B	always
ZBOU(I,IS)	Z_i^s	V.B.2	IATRTE(IS) = 4
ZROBSF	same	V.C.1	always

(continued)

Table VII-1. (Concluded)

Notes:

1. FORTRAN inputs marked with an asterisk also appear on Table VII-2.
2. The indices used for the FORTRAN inputs on this table are as follows: IS denotes side; I denotes either weapon type or aircraft type (as appropriate for the input in question) on side IS, and J denotes weapon type on the enemy's side.
3. The indices used in the algebraic notation on the table are as follows: s denotes side, i denotes weapon type on side s in non-underlined expressions, i denotes aircraft type on side s in underlined expressions, and j denotes weapon type on the enemy's side.

Table VII-2. Definitions of FILEIN, IALMTH, IALVAL, IALWT, IALWTA, IATRTE, IFVUL, INVAL, IRSCRN, IVSUMP, IWSCRN, IWSCRO, NAMVAR, NPER, and NWNAME

FILEIN = the name of the file from which other inputs are to be read. The code always asks for the input name for FILEIN to be typed from the keyboard, without quote marks around it. If no extension is given, the code automatically assumes an extension of ".DAT". FILEIN is read in CHARACTER*12 format.

IALMTH(IS) = the index for the method to be used to compute allocations of fire for side IS, where:

0 = allocations are proportional to the weighted number of enemy weapons of each type,

1 = allocations are proportional to X(I,J)*(the weighted number of enemy weapons of type J),

2 = allocations are a convex combination of:

a) all type-I weapons on side IS shoot at the type-J enemy woapon that maximizes X(I,J), and

b) allocations that are proportional to X(I,J)*(the weighted number of enemy weapons of type J), and

3 = allocations are a convex combination of:

a) all type-I weapons on side IS shoot at the type-J enemy weapon that maximizes X(I,J), and

b) allocations that are proportional to the weighted number of enemy weapons of each type,

where if IALVAL(IS) = 1 then

X(I,J) = E(I,IS)*P(I,J,IS)*E(J,JS)*P(J,I,JS),

and if IALVAL(IS) > 1 then

X(I,J) = E(I,IS)*P(I,J,IS)*V(J,JS),

where JS = 3-IS and V(J,JS) is the score (value) of an enemy weapon of type J computed using the method specified by the input INVAL.

IALVAL(IS) = the index for the relationship between allocations of fire and weapons scores (values) to be used for side IS, where:

1 = allocations are independent of weapons scores,

2 = allocations are interconnected with weapons scores, and

3 = allocations depend on base case (INVAL = 0) or first time-period (INVAL > 0) scores.

IALWT = the index for whether certain inputs are required to compute allcoations of fire for fully-interacting weapons, where:

0 = data for the input arrays TYPCLA and TYPCLW are not to be entered, the code calculates allocations of fire as if TYPCLA(I,J,IS) = 1/N(JS) and TYPCLW(I,IS) = 1 for all relevant I and J on both sides, and

1 = data for the input arrays TYPCLA and TYPCLW must be entered.

(continued)

Table VII-2 (Continued)

IALWTA = the index for whether a certain input is required to compute allcoations of fire for (non-interacting) aircraft, where:
0 = data for the input array TPCLAA are not to be entered, the code calculates allocations of fire as if TPCLAA(I,J,IS) = 1/N(JS) for all relevant I and J on both sides, and
1 = data for the input array TPCLAA must be entered.

IATRTE(IS) = the index for the method to be used to determine the attrition inflicted by weapons on side IS, where this attrition is computed by assuming:
0 = a Lanchester square firing process,
1 = a perfect shoot-look-shoot firing process,
2 = a uniformly coordinated firing process,
3 = an uncoordinated firing process, and
4 = a mixture of uniformly coordinated and uncoordinated fires.

IFVUL = the index for whether a certain input is required to compute the fraction of weapons that are vulnerable to enemy fire, where:
0 = data for the input array U are not to be entered, the code calculates vulnerability as if U(I,IS) = 1 for all relevant I on side IS, and
1 = data for the input array U must be entered.

INVAL = the index for the method(s) to be used to compute weapon scores (i.e., weapon values), where:
0 = Specified,
1 = APP,
2 = APPVUL,
3 = PEXPOT,
4 = DYNPOT,
5 = LEVPOT, and
6 = all methods.

IRSCRN = the index for whether certain inputs are to be entered interactively (i.e., read from the keyboard) when running COMBAT, where:
0 = no (only FILEIN is read from the keyboard, all other data are contained in the input data file), and
1 = yes.

IVSUMP = the index for the scoring method to be used when INVAL = 6 to determine the arrays that are displayed on the summary output table, where:
1 = APP,
2 = APPVUL,
3 = PEXPOT,
4 = DYNPOT, and
5 = LEVPOT.

(continued)

Table VII-2. (Concluded)

IWSCRN = the index for whether certain results will be written to the screen in addition to being written to an output file, where:
0 = no,
1 = yes, and
2 = base this decision on IRSCRN and IWSCRO--if IRSCRN = 1 and IWSCRO = 1, then yes; otherwise no.

IWSCRO: If both IRSCRN = 1 and IWSCRN = 2, then IWSCRO determines whether certain results will be written to the screen in addition to being written to an output file, where:
0 = no, and
1 = yes.
If either IRSCRN ≠ 1 or IWSCRN ≠ 2, then IWSCRO is not used.

NAMVAR = an input (in CHARACTER*13 format) that must be provided as the first entry on each record of the input data file. The code makes no use whatsoever of this input--its sole purpose is to help users of COMBAT identify the other data entries on that record. This input must have single quote marks around it. The null input, '', is allowed for NAMVAR.

NPER = the number of time periods over which combat attrition is to be simulated. Setting NPER = 0 is useful for comparing all of the methods for computing weapons scores (INVAL = 6) based on the initial (input) data. If INVAL = 6 and NPER ≥ 1, then the results displayed on the ".SUM" output file apply to values computed after NPER periods of attrition have been assessed. If INVAL ≤ 5 and NPER = 0, then the ".OUT" output file will only display the inputs and some intermediate calculated arrays. Major results are displayed on the ".OUT" file only if NPER ≥ 1.

NWNAME = an input (in CHARACTER*8 format) that will be requested from the keyboard (without quote marks around it) if IRSCRN = 1 in order to name the output file(s). If no entry is provided (i.e., the response to this request is just the return (or enter) keystroke), then the (first) name of FILEIN will be used to name the output file(s); otherwise, this entry will be used as the first name of the output file(s). (If IRSCRN = 0, the first name of FILEIN will always be used to name the output file(s).)

Table VII-3. Definitions of IALMTH, IALVAL, and INVAL

IALMTH(IS) = the index for the method to be used to compute allocations of fire for side IS, where:

0 = allocations are proportional to the weighted number of enemy weapons of each type,

1 = allocations are proportional to X(I,J)*(the weighted number of enemy weapons of type J),

2 = allocations are a convex combination of:

a) all type-I weapons on side IS shoot at the type-J enemy weapon that maximizes X(I,J), and

b) allocations that are proportional to X(I,J)*(the weighted number of enemy weapons of type J), and

3 = allocations are a convex combination of:

a) all type-I weapons on side IS shoot at the type-J enemy weapon that maximizes X(I,J), and

b) allocations that are proportional to the weighted number of enemy weapons of each type,

where if IALVAL(IS) = 1 then

X(I,J) = E(I,IS)*P(I,J,IS)*E(J,JS)*P(J,I,JS),

and if IALVAL(IS) > 1 then

X(I,J) = E(I,IS)*P(I,J,IS)*V(J,JS),

where JS = 3-IS and V(J,JS) is the score (value) of an enemy weapon of type J computed using the method specified by the input INVAL.

IALVAL(IS) = the index for the relationship between allocations of fire and weapons scores (values) to be used for side IS, where:

1 = allocations are independent of weapons scores,

2 = allocations are interconnected with weapons scores, and

3 = allocations depend on base case (INVAL = 0) or first time-period (INVAL > 0) scores.

INVAL = the index for the method(s) to be used to compute weapon scores (i.e., weapon values), where:

0 = Specified,
1 = APP,
2 = APPVUL,
3 = PEXPOT,
4 = DYNPOT,
5 = LEVPOT, and
6 = all methods.

Table VII-4 gives the range of allowable values for each of the inputs to COMBAT. These inputs are listed in alphabetical order on Table VII-4. One of these inputs, IALVAL(IS), has a particular set of restrictions concerning its allowable values. These restrictions are listed on Table VII-5.

2. Input Files and Procedures

COMBAT contains a rather rudimentary input routine. Inputs must be entered in a certain order, discussed below, with some (at least one) of these inputs being provided from the keyboard, and others being provided from an input data file. The first input that COMBAT expects, FILEIN, is the name of that input data file, and COMBAT always expects this name to be provided from the keyboard. The input name for FILEIN can be entered either with or without an extension. If no extension is given, the default extension of ".DAT" is appended to the input name. The next inputs that COMBAT expects are values for NAMVAR, IRSCRN, and IWSCRN, which it expects to find as the first set of entries in the input data file specified by FILEIN.

NAMVAR is a character variable that is read in as the first entry of each record in the input data file. The value of NAMVAR is totally ignored by the code--its sole purpose is to help users of COMBAT identify the other entries in each record.

If the value given IRSCRN is zero, then COMBAT will expect all of the rest of the inputs to come from the input data file (and so, in this case, all of the inputs to COMBAT except for FILEIN must be in the input data file). The order in which these inputs are to appear in the input data file when IRSCRN = 0 is given on Table VII-6. (Table VII-6 also repeats the conditions under which each of these inputs is required.) Figure VII-1 gives an example of an input data file with IRSCRN = 0.

If the value given IRSCRN is one, then COMBAT will expect some inputs (in addition to FILEIN) to come from the keyboard and others to be in the input data file. The inputs that COMBAT expects from the keyboard when IRSCRN = 1 (in the order in which they are requrested) are given on Table VII-7. The order in which inputs are to appear in the input data file where IRSCRN = 1 is given on Table VII-8. (Table VII-7 also gives the conditions under which each of its inputs is requested, and Table VII-8 also repeats the conditions under which each of its inputs is required.) Figure VII-2 gives an example of an input data file with IRSCRN = 1.

Except for FILEIN, all inputs are read in as list-directed records. FILEIN is read in A12 format.

Table VII-4. Range of Allowable Values of Inputs

Input	Range	Input	Range
AC(I,IS)	[0,∞)	N(IS)	{1,2,...,NTYPE}[1]
CAREAI	[0,∞)	NA(IS)	{1,2,...,NATYPE}[1]
E(I,IS)	[0,∞)	NAMVAR	CHARACTER*13
EA(I,IS)	[0,∞)	NPER	{0,1,2,...}
FILEIN	CHARACTER*12	NWNAME	CHARACTER*8
IALMTH(IS)	{0,1,2,3}	P(I,J,IS)	[0,1]
IALVAL(IS)	{1,2,3}	PA(I,J,IS)	[0,1]
IALWT	{0,1}	Q(I,IS)	[0,1]
IALWTA	{0,1}	QA(I,IS)	[0,1]
IATRTE(IS)	{0,1,2,3,4}	TPCLAA(I,J,IS)	[0,1]
IFVUL	{0,1}	TYPCLA(I,J,IS)	[0,1]
INVAL	{0,1,2,3,4,5,6}	TYPCLW(I,IS)	(0,∞)
IRSCRN	{0,1	U(I,IS)	[0,1]
IVSUMP	{1,2,3,4,5}	V(I,IS)	[0,∞)
IWSCRN	{0,1,2}	W(I,IS)	[0,∞)
IWSCPO	{0,1}	ZBOU(I,IS)	[0,1]
I1	{1,2,...,N(1)}	ZROBSF	[0,1] ∪ [2,4]

[1] NTYPE and NATYPE are symbolic constants whose vlaues are set in a PARAMETER statement--see Section B for details.

Table VII-5. Special Restrictions that Apply to IALVAL(IS)

Both of the following restrictions apply to the values input for IALVAL(IS).

(1) If IALMTH(1) = 0, then IALVAL(1) must equal 1; and if IALMTH(2) = 0, then IALVAL(2) must equal 1.

(2) IALVAL(1) must equal IALVAL(2) unless INVAL = 0. If INVAL = 0 then:

(a) IALVAL(1) = 1 means that IALVAL(2) must equal 1 or 3,

(b) IALVAL(1) = 2 means that IALVAL(2) must equal 2, and

(c) IALVAL(1) = 3 means that IALVAL(2) must equal 1 or 3.

Table VII-6. File Inputs, In Order, If IRSCRN = 0

Input	Conditions Under Which Input is Required
'NAMVAR', IRSCRN, IWSCRN	always
'NAMVAR', INVAL, NPER	always
'NAMVAR', IALVAL(1), IALVAL(2)	always
'NAMVAR', IALMTH(1), IALMTH(2)	always
'NAMVAR', IVSUMP	INVAL = 6
'NAMVAR', N(1), N(2), I1	always
'NAMVAR', V(I,1)	INVAL = 0
'NAMVAR', V(I,2)	"
'NAMVAR', Q(I,1)	IALMTH(1) = 2 or 3
'NAMVAR', Q(I,2)	IALMTH(2) = 2 or 3
'NAMVAR', W(I,1)	always
'NAMVAR', W(I,2)	"
'NAMVAR', E(I,1)	always
'NAMVAR', E(I,2)	"
'NAMVAR', P(1,J,1)	always
'NAMVAR', .	"
'NAMVAR', .	"
'NAMVAR', .	"
'NAMVAR', P(N(1),J,1)	"
'NAMVAR', P(1,J,2)	"
'NAMVAR', .	"
'NAMVAR', .	"
'NAMVAR', .	"
'NAMVAR', P(N(2),J,2)	"
'NAMVAR', ZROBSF, CAREA	always
'NAMVAR', IATRTE(1), IATRTE(2)	always
'NAMVAR', ZBOU(I,1)	IATRTE(1) = 4
'NAMVAR', ZBOU(I,2)	IATRTE(2) = 4
'NAMVAR', IALWT, IFVUL	always

(continued)

Table VII-6. (Continued)

Input	Conditions Under Which Input is Required
'NAMVAR', TYPCLW(I,1)	IALWT = 1
'NAMVAR', TYPCLW(I,2)	"
'NAMVAR', TYPCLA(1,J,1)	IALWT = 1
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', TYPCLA(N(1),J,1)	"
'NAMVAR', TYPCLA(1,J,2)	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', TYPCLA(N(2),J,2)	"
'NAMVAR', U(I,1)	IFVUL = 1
'NAMVAR', U(I,2)	"
'NAMVAR', NA(1), NA(2)	always
'NAMVAR', QA(I,1)	NA(•) ≥ 1 AND IALMTH(1) = 2 or 3
'NAMVAR', QA(I,2)	NA(•) ≥ 1 AND IALMTH(2) = 2 or 3
'NAMVAR', AC(I,1)	NA(•) ≥ 1
'NAMVAR', AC(I,2)	"
'NAMVAR', EA(I,1)	NA(•) ≥ 1
'NAMVAR', EA(I,2)	"
'NAMVAR', PA(1,J,1)	NA(•) ≥ 1
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', PA(NA(1),J,1)	"
'NAMVAR', PA(1,J,2)	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', PA(NA(2),J,2)	"
'NAMVAR', IALWTA	NA(•) ≥ 1

(continued)

Table VII-6. (Concluded)

Input	Conditions Under Which Input is Required
'NAMVAR', TPCLAA(1,J,1)	NA(•) ≥ 1 and IALWTA = 1
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', TPCLAA(NA(1),J,1)	"
'NAMVAR', TPCLAA(1,J,2)	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', TPCLAA(NA(2),J,2)	"

```
'IRSCRN,IWSCRN' 0 0
'INVAL,NPER    ' 6 0
'IALVAL(IS)    ' 1 1
'IALMTH(IS)    ' 0 0
'IVSUMP        ' 1
'N(1),N(2),I1  ' 4 5 1
'W(.,1)        ' 150 75 100 50
'W(.,2)        ' 250 100 75 100 150
'E(.,1)        ' .8 .4 .6 1
'E(.,2)        ' .8 .2 .8 .4 .8
'P(1,.,1)      ' .3 .4 .45 .5 .55
'P(2,.,1)      ' .25 .3 .35 .4 .45
'P(3,.,1)      ' .2 .25 .3 .35 .4
'P(4,.,1)      ' .15 .2 .25 .3 .35
'P(1,.,2)      ' .15 .25 .3 .35
'P(2,.,2)      ' .1 .2 .25 .3
'P(3,.,2)      ' .08 .15 .2 .25
'P(4,.,2)      ' .06 .1 .15 .2
'P(5,.,2)      ' .04 .05 .1 .15
'ZROBSF,CAREA  ' 3. 2.
'IATRTE(IS)    ' 0 0
'IALWT,IFVUL   ' 0 0
'NA(1),NA(2)   ' 3 4
'AC(.,1)       ' 150 75 100
'AC(.,2)       ' 250 100 75 125
'EA(.,1)       ' 1.8 2.8 3.3
'EA(.,2)       ' 0.8 1.4 2.4 3.8
'PA(1,.,1)     ' .3 .4 .45 .3 .4
'PA(2,.,1)     ' .25 .3 .35 .25 .35
'PA(3,.,1)     ' .2 .25 .3 .2 .3
'PA(1,.,2)     ' .15 .25 .3 .4
'PA(2,.,2)     ' .1 .2 .25 .35
'PA(3,.,2)     ' .08 .15 .2 .3
'PA(4,.,2)     ' .15 .25 .3 .4
'IALWTA        ' 0
```

Figure VII-1. A Sample Input Data File with IRSCRN = 0

Table VII-7. Keyboard Inputs (In the Order In Which They Can be Requested) If IRSCRN = 1

Input	Conditions Under Which Input is Requested
FILEIN	always
NWNAME	always
INVAL	always
IVSUMP	INVAL = 6
IALVAL(1),	always
IALVAL(2)	"
IALMTH(1)	always
IALMTH(2)	always
NPER	always
V(I,1)	INVAL = 0
V(I,2)	INVAL = 0
Q(I,1)	IALMTH(1) = 2 or 3
Q(I,2)	IALMTH(2) = 2 or 3
IWSCRO	IWSCRN = 2

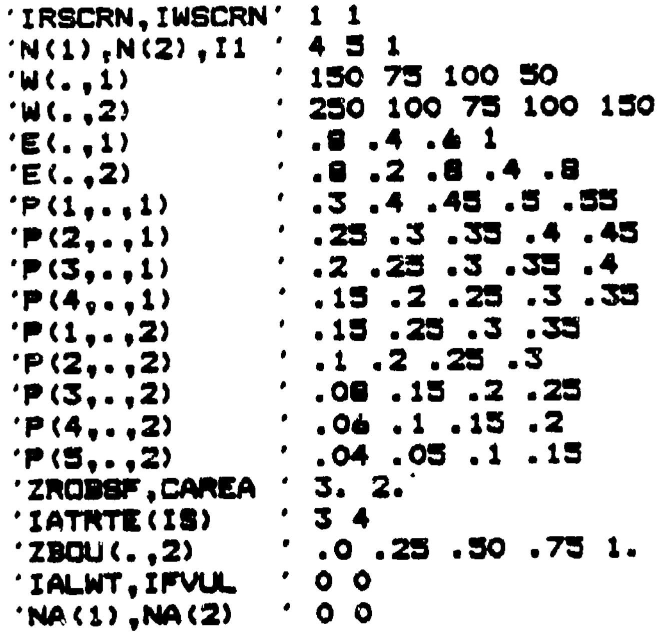
Table VII-8. File Inputs, In Order, If IRSCRN = 1

Input	Conditions Under Which Input is Required
'NAMVAR', IRSCRN, IWSCRN	always
'NAMVAR', N(1), N(2), I1	always
'NAMVAR', W(I,1)	always
'NAMVAR', W(I,2)	"
'NAMVAR', E(I,1)	always
'NAMVAR', E(I,2)	"
'NAMVAR', P(1,J,1)	always
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', P(N(1),J,1)	"
'NAMVAR', P(1,J,2)	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', P(N(2),J,2)	"
'NAMVAR', ZROBSF, CAREA	always
'NAMVAR', IATRTE(1), IATRTE(2)	always
'NAMVAR', ZBOU(I,1)	IATRTE(1) = 4
'NAMVAR', ZBOU(I,2)	IATRTE(2) = 4
'NAMVAR', IALWT, IFVUL	always
'NAMVAR', TYPCLW(I,1)	IALWT = 1
'NAMVAR', TYPCLW(I,2)	"
'NAMVAR', TYPCLA(1,J,1)	IALWT = 1
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', TYPCLA(N(1),J,1)	"
'NAMVAR', TYPCLA(1,J,2)	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', TYPCLA(N(2),J,2)	"

(continued)

Table VII-8. (Concluded)

Input	Conditions Under Which Input is Required
'NAMVAR', U(I,1)	IFVUL = 1
'NAMVAR', U(I,2)	"
'NAMVAR', NA(1), NA(2)	always
'NAMVAR', QA(I,1)	NA(•) ≥ 1 AND IALMTH(1) = 2 or 3
'NAMVAR', QA(I,2)	NA(•) ≥ 1 AND IALMTH(2) = 2 or 3
'NAMVAR', AC(I,1)	NA(•) ≥ 1
'NAMVAR', AC(I,2)	"
'NAMVAR', EA(I,1)	NA(•) ≥ 1
'NAMVAR', EA(I,2)	"
'NAMVAR', PA(1,J,1)	NA(•) ≥ 1
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', PA(NA(1),J,1)	"
'NAMVAR', PA(1,J,2)	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', PA(NA(2),J,2)	"
'NAMVAR', IALWTA	NA(•) ≥ 1
'NAMVAR', TPCLAA(1,J,1)	NA(•) ≥ 1 and IALWTA = 1
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', TPCLAA(NA(1),J,1)	"
'NAMVAR', TPCLAA(1,J,2)	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', •	"
'NAMVAR', TPCLAA(NA(2),J,2)	"



```
'IRSCRN,IWSCRN'  1 1
'N(1),N(2),I1  '  4 5 1
'W(.,1)         '  150 75 100 50
'W(.,2)         '  250 100 75 100 150
'E(.,1)         '  .8 .4 .6 1
'E(.,2)         '  .8 .2 .8 .4 .8
'P(1,.,1)       '  .3 .4 .45 .5 .55
'P(2,.,1)       '  .25 .3 .35 .4 .45
'P(3,.,1)       '  .2 .25 .3 .35 .4
'P(4,.,1)       '  .15 .2 .25 .3 .35
'P(1,.,2)       '  .15 .25 .3 .35
'P(2,.,2)       '  .1 .2 .25 .3
'P(3,.,2)       '  .08 .15 .2 .25
'P(4,.,2)       '  .06 .1 .15 .2
'P(5,.,2)       '  .04 .05 .1 .15
'ZROBSF,CAREA   '  3. 2.
'IATRTE(IS)     '  3 4
'ZBOU(.,2)      '  .0 .25 .50 .75 1.
'IALWT,IFVUL    '  0 0
'NA(1),NA(2)    '  0 0
```

Figure VII-2. A Sample Input Data File with IRSCRN = 1

B. SYMBOLIC CONSTANTS

COMBAT defines and uses the symbolic constant NTYPE to declare the appropriate dimension bounds of arrays that are functions of the number of types of (fully interacting) weapons on either side, and it defines and uses the symbolic constant NATYPE in an analogous manner for (non-interacting) aircraft. Values for NTYPE and NATYPE are set by the PARAMETER statement contained in the file COMBAT.PAR. As indicated on Table VII-4, N(IS) should not exceed NTYPE and NA(IS) should not exceed NATYPE for either value of IS. A user of COMBAT can freely change the values of NTYPE and NATYPE by editing the file COMBAT.PAR, recompiling the program units that include COMBAT.PAR, and relinking. The internal calculations made by COMBAT are designed to handle arbitrarily large values for NTYPE and NATYPE--the limits here are essentially the size of available computer memory and practical running times. However, while some of the output formats can handle large values for NTYPE and NATYPE and produce easily readable results, others will produce output that is difficult to read.

The file COMBAT.PAR also sets the symbolic constant MOEDIM = 5. This constant is fixed at 5 in that it cannot be meaningfully changed without also changing a significant number of executable statements in various portions of the code.

The symbolic constant MAXNWC is set and used in Subroutine PSLSAE, which is called to compute shoot-look-shoot attrition if IATRTE(IS) = 1 for side IS. If IATRTE(IS) = 1, then MAXNWC must be large enough so that

$$\sum_{I=1}^{NS(IS)} W(I,IS) / CAREA \leq MAXNWC .$$

MAXNWC is only used to declare dimension bounds of some local arrays in Subroutine PSLSAE, and it can be set equal to one if shoot-look-shoot attrition is not being simulated. Only the size of available memory and practical running times limit how large MAXNWC can be.

C. OUTPUTS

1. Output Files

All relevant inputs, selected intermediate results, and major results concerning weapons scores, allocations of fire, and attrition produced by a run of COMBAT are written onto a file with the extension ".OUT". If IRSCRN = 0, then COMBAT sets the first name of this output file to be the same as the first name of the input data file (which is stored in FILEIN). If IRSCRN = 1, then COMBAT requests that a first name for this output file be entered from the keyboard. (This is the entry for the character input NWNAME.) If, in response to this request, no entry is given (i.e., only the return or enter key is struck), then COMBAT again uses the first name of the input data file here. Otherwise, if a meaningful entry is given in response to this request, then this entry is used as the first name of this output file.

If INVAL = 6, then COMBAT also writes selected inputs and results concerning weapons scores as computed by each of the methods (APP, APPVUL, PEXPOT, LEVPOT, and DYNPOT) onto a file with the extension ".SUM". The first name of this file will always be the same as the first name of the ".OUT" file produced by that run.

Both of these output files are opened with STATUS = 'UNKNOWN'.

2. Output Data

a. Output of Inputs

As indicated above, the values of all relevant inputs are written onto the corresponding ".OUT" file, and (if INVAL = 6) a selected subset of these input values are also written onto the ".SUM" file. The values of these inputs on these files are listed following their FORTRAN names, and the definitions of these FORTRAN names can be found using Table VII-1 above.

b. Some Calculated Arrays

After listing inputs, selected intermediate results are listed on the ".OUT" file (if NPER > 0) and on the ".SUM" file (if INVAL = 6). These results are called "calculated arrays" on those files. A list (in alphabetical order by FORTRAN name) of all of these calculated arrays is given on Table VII-9. That table also defines these arrays in terms of the algebraic notation introduced in Chapters II and III above. Descriptions of these arrays (in a logical order) are given below.

Table VII-9. Calculated Arrays Written to Output Files

FORTR⁄ . N Array Name	Equivalent Term in Algebraic Notation	Chapter & Section Defining Relevant Algebraic Notation
A(I,J,IS)	A^{s}_{ij}	II.B
AA(I,J,IS)	$\dfrac{\hat{\underline{A}}^{s}_{ij} / \hat{W}^{s'}_{j}}{\sum_{j'=1}^{N^{s'}} \hat{\underline{A}}^{s}_{ij'} / \hat{W}^{s'}_{j'}}$	III.B
AB(I,J,IS)	$\dfrac{\hat{A}^{s}_{ij} / \hat{W}^{s'}_{j}}{\sum_{j'=1}^{N^{s'}} \hat{A}^{s}_{ij'} / \hat{W}^{s'}_{j'}}$	III.B
ABI(I,J,IS)	$\dfrac{A^{s}_{ij} / W^{s'}_{j}}{\sum_{j'=1}^{N^{s'}} A^{s}_{ij'} / W^{s'}_{j'}}$	II.B
AK(I,J,IS)	$\underline{K}^{s}_{ij}$	II.D.2.a
ALA(I,J,IS)	$\underline{A}^{s}_{ij}$	II.D.2.a
FRK(I,J,IS)	$\hat{K}^{s}_{ij}$	II.C
RK(I,J,IS)	K^{s}_{ij}	II.B
RKK(I,K,IS)	$\sum_{j=1}^{N^{s'}} K^{s}_{ij} K^{s'}_{jk}$	II.B

A(I,J,IS): This aray gives the operational allocation of fire for the (fully interacting) weapons in the particular forces in question. That is, A(I,J,IS) is the FORTRAN notation for the algebraic term A^s_{ij} which is defined in Section B of Chapter II, is calculated as described in Chapters III and IV, and is used extensively throughout COMBAT.

AB(I,J,IS): Based (optionally, if IALWT = 1) on the input allocation of fire TYPCLA against the enemy force given by TYPCLW, this array gives what the allocation of fire (of fully interacting shooters) would be against a hypothetical enemy force that consisted of equal numbers of all of the types of (fully interacting enemy) weapons being considered. This array is useful for helping understand the impact of particular values for TYPCLA and TYPCLW, for comparing allocations of fire with those obtained using other values for TYPCLA and TYPCLW, for comparing allocations of fire when IALMTH(IS) ≥ 1, and for constructing hypothetical examples. (If IALWT = 0, Subroutine INPUT sets AB(I,J,IS) = 1/N(JS) for all relevant I.) This array is also used by Subroutine ALLOCT to help compute A(I,J,IS).

ABI(I,J,IS): If IALMTH(IS) = 0 and IFVUL = 0, then the operational allocation of fire being used, A, is computed directly from AB and W, and ABI(I,J,IS) = AB(I,J,IS) for all relevant I and J for both sides. If IALMTH(IS) ≥ 1 and/or IFVUL = 1, then A is computed based, in part, on other terms (see Chapters II, III, and IV above for details). In this latter case, the array ABI is what the array AB would have needed to have been in order to have computed the same values for A based solely on AB and W (i.e., if IALMTH and IFVUL had been zero).

RK(I,J,IS): This array gives the measures of the capabilities of (fully interacting) weapons to kill enemy weapons that are used by COMBAT to compute weapon scores. That is, RK(I,J,IS) is the FORTRAN notation for the algebraic term K^s_{ij} which is defined in Section B of Chapter II and is used throughout that chapter in the computation of these scores.

RKK(I,K,IS): For each IS, this is the array whose non-negative eigenvector yields the antipotential potential weapon scores for (fully interacting) weapons on side IS. That is, RKK(I,K,IS) is the FORTRAN notation for the matrix product

$$\sum_{j=1}^{N^{s'}} K^{s}_{ij} K^{s'}_{jk}$$

where K^{s}_{ij} is as defined in Section B of Chapter II. In terms of the notation used in Appendix A, RKK(i,i',1) corresponds to $\overline{K}_{ii'}$.

FRK(I,J,IS): This array gives a measure of the rate at which weapons of type J can be killed by weapons of type I on side IS. In particular, FRK(I,J,IS) is the FORTRAN notation for the algebraic term $\hat{K}^{s}_{ij}$ which is defined in Section C of Chapter II and is used in Sections 2 through 5 of that chapter to address vulnerability in the calculation of weapons scores.

ALA(I,J,IS): This array is the equivalent of A(I,J,IS) for non-vulnerable weapons. That is, ALA(I,J,IS) is the FORTRAN notation for the algebraic term $\underset{\sim}{A}^{s}_{ij}$ as aefined in Section D.2.a of Chapter II. This array is not computed, used, or displayed if NA(IS) = 0.

AA(I,J,IS): This array is the equivalent of array AB for non-vulnerable weapons. It is not computed, used, or displayed if NA(IS) = 0.

AK(I,J,IS): This aray is the equivalent of RK(I,J,IS) for non-vulnerable weapons. That is, AK(I,J,IS) is the FORTRAN notation for the algebraic term $\underset{\sim}{K}^{s}_{ij}$ as defined in Section D.2.a of Chapter II. This aray is not computed, used, or displayed if NA(IS) = 0.

c. Major Results Displayed on the ".OUT" File

After displaying the inputs and the calculated arrays AB and, if NA > 0, AA, the ".OUT" file displays results time period by time period. (If NPER = 0, only the inputs are displayed; no calculated arays or results are given.) For each time period simulated, the ".OUT" file displays some calculated arrays followed by some major results for that time period. These major results are as follows.

First, results concerning side 1 firing at side 2 are given. For each type of weapon on side 2, these results give the number of weapons at the start of the time period, the number killed during that time period, the number remaining at the end of the time period, and the score of that type of weapon for that time period. Next, a cumulative killer-victim scoreboard is displayed. Each row of this scoreboard corresponds to a particular type of (shooting) weapon on side 1, except for the last row which gives a column total. Each column of this scoreboard corresponds to a particular type of (target) weapon on side 2, except for the last column which corresponds to the resulting score killed (i.e., the sum over weapon types of number of weapons killed times the score of those weapons). Before discussing such details concerning the calculation of this scoreboard, note that all of the corresponding results concerning side 2 firing at side 1 are displayed next. That is, after displaying the side 1 firing at side 2 killer-victim scoreboard, the ".OUT" file displays the initial numbers of weapons by type on side 1, the numbers of those weapons that are killed during the time period, the resulting numbers of those weapons remaining, and the scores of those weapons for that time period, followed by a cumulative killer-victim scoreboard for side 2 firing at side 1.

In terms of the algebraic notation used in Chapters II through V above, some details concerning the calculations of these killer-victim scoreboards are as follows. Let C^s_{ij} denote the number of kills of weapons of type j to be credited to shooting weapons of type i on side s during the time period in question. Subroutine KVSCRB computes values for C^s_{ij} using the formula

$$C^s_{ij} = D^{s'}_j (W^s_i K^s_{ij} / \sum_{i'=1}^{N^s} W^s_{i'} K^s_{i'j})$$

where D is as defined in Section C.1 of Chapter V and the other terms are as defined in Section B of Chapter II. The time-period increment for the last column of this scoreboard is computed as

$$\sum_{j=1}^{N^{s'}} C^s_{ij} V^{s'}_j ,$$

and the time period increment for the last row of this scoreboard is simply

$$\sum_{i=1}^{N^s} C^s_{ij} = D^{s'}_j .$$

See Section B.4.g of Chapter III of Reference [26] and (all of) Reference [29] for general discussion of the computation of killer-victim scoreboards in models, like COMBAT, that calculate weapon losses using attrition equations.

The last set of results displayed on the ".OUT" file for the time period in question is a set of force comparisons. These force comparisons give the initial (at the start of the time period) force ratio, the final (at the end of the time period) force ratio, and a set of initial, final, and cumulative final force comparisons as described in Section E of Chapter II. (For example, the entries under "–/minimum" correspond to values computed using the function g_n as defined in that section.

While the ".OUT" file displays all relevant inputs, currently no results concerning non-vulnerable weapons are given on that file. Such output could easily be added to that file, if desired.

d. Major Results Displayed on the ".SUM" File

If NPER = 0, the results displayed on the ".SUM" file are based diectly on the input data. If NPER ≥ 1, all of the results displayed on that file are those that apply after NPER time periods of attrition have been assessed. The ".SUM" file first displays a few relevant inputs (including the time period, NPER), followed by some calculated arrays (defined in Section b above), followed by selected major results. These major results are as follows.

First, the weapons scores and resulting force strengths as computed by each of the methods discussed in Chapter II are displayed for both sides. These scores are scaled so that weapon type I1 on side 1 receives a score of 1 (I1 is an input). Next, the relevant values for β are displayed (the FORTRAN variable BETA corresponds to β as defined for each of the methods in Chapter II). Next displayed are the corresponding weapons scores and resulting force strengths scaled so that the average score of all of the (fully interacting) weapons equals 1. The rationale for this rescaling is discussed in Section C.6 of Chapter II.

Following these rescaled scores and strengths, a set of force comparisons is displayed. These comparisons give, for each scoring method, the side 1 over side 2 force ratio, its inverse, and its inverse to the power P where P coresppnds to ρ as defined and discussed in Appendix E, below. See that appendix for the rationale behind the selection of values for P. These comparisons also give, for each scoring method, the values of the force comparison measures g_n, g_h, g_g, and g_t as defined and discussed in Section E of Chapter II, above.

If all of the weapons systems being considered are fully interacting weapons (i.e., NA = 0), then this set of force comparisons concludes the display of results on the ".SUM" file. If non-vulnerable weapons systems are being addressed (NA > 0), then the results displayed so far on this file concern only fully interacting weapons, and additional results that concern non-vulnerable weapons are then displayed. In particular, scores and strengths for these non-vulnerable systems are given for each scoring method.

This file then concludes with a corresponding set of force comparisons whose values are based on the scores and strengths of both the fully interacting and the non-vulnerable weapons systems.

D. SOURCE CODE

The source code for COMBAT is contained in 28 files. Twenty-one of these files contain program units, six contain COMMON blocks, and one contains a PARAMETER statement.

COMBAT is comprised of 21 program units and each program unit is contained in a separate file with the same (first) name as the symbolic name of that program unit and with an extension of ".FOR" (e.g., Subroutine INPUT is contained in file INPUT.FOR, and it is the only entry in that file). The names of these files are listed in alphabetical order on Table VII-10.

All of the COMMON blocks in COMBAT are labeled COMMON blocks and all are contained in files with the extension ".CMN." There are six such files, only one of which (COMBAT.CMN) contains more than one COMMON block. Table VII-10 also lists these files in alphabetical order. The remaining source code file, COMBAT.PAR, contains the parameter statement discussed in Section C. Many of the program units of COMBAT contain INCLUDE statements that involve COMBAT.PAR and one or more of the COMMON block files.

Figure VII-3 gives a calling tree for the program units of COMBAT.

Finally, Table VII-11 gives the size of each program unit in terms of the number of initial lines (other than comments), the number of continuation lines, the number of "real" comment lines, and the number of essentially blank comment lines. For the purposes of Table VII-11, a comment line is called "blank" if it is blank in columns 7 through 70 inclusive, and it is called "real" otherwise. Also, an included file is counted as one initial line in Table VII-11, no matter how many lines are in that file. The resulting total size of COMBAT is given at the end of Table VII-10.

Table VII-10. COMBAT Source Code Files

Program Unit Files

File	Program Unit
AIRVAL.FOR	Subroutine AIRVAL
ALLOC.FOR	Subroutine ALLOC
ALLOCA.FOR	Subroutine ALLOCA
ATTRIT.FOR	Subroutine ATTRIT
BINOAE.FOR	Subroutine BINOAE
BINOMP.FOR	Subroutine BINOMP
CALWT.FOR	Subroutine CALWT
CALWTA.FOR	Subroutine CALWTA
COMBAT.FOR	Program COMBAT
CWTALA.FOR	Subroutine CWTALA
CWTDAL.POR	Subroutine CWTDAL
DATOB.FOR	Function DATOB
INPUT.FOR	Subroutine INPUT
KVSCRB.FOR	Subroutine KVSCRB
PSLSAE.FOR	Subroutine PSLSAE
RELNEF.FOR	Subroutine RELNEF
UNIBIN.FOR	Subroutine UNIBIN
UNIFAE.FOR	Subroutine UNIFAE
VALUE.FOR	Subroutine VALUE
WRTOUT.FOR	Subroutine WRTOUT
WRTSUM.FOR	Subroutine WRTSUM

COMMON Block Files

File	COMMON Blocks
ALLOUT.CMN	ALLOUT
COMBAT.CMN	CMBTAI CMBTJZ
COMOUT.CMN	COMOUT
IOUNIT.CMN	IOUNIT
SUMOUT.CMN	SUMOUT
VALUWK.CMN	VALUWK

Other Files

File	Description
COMBAT.PAR	Parameter Statement

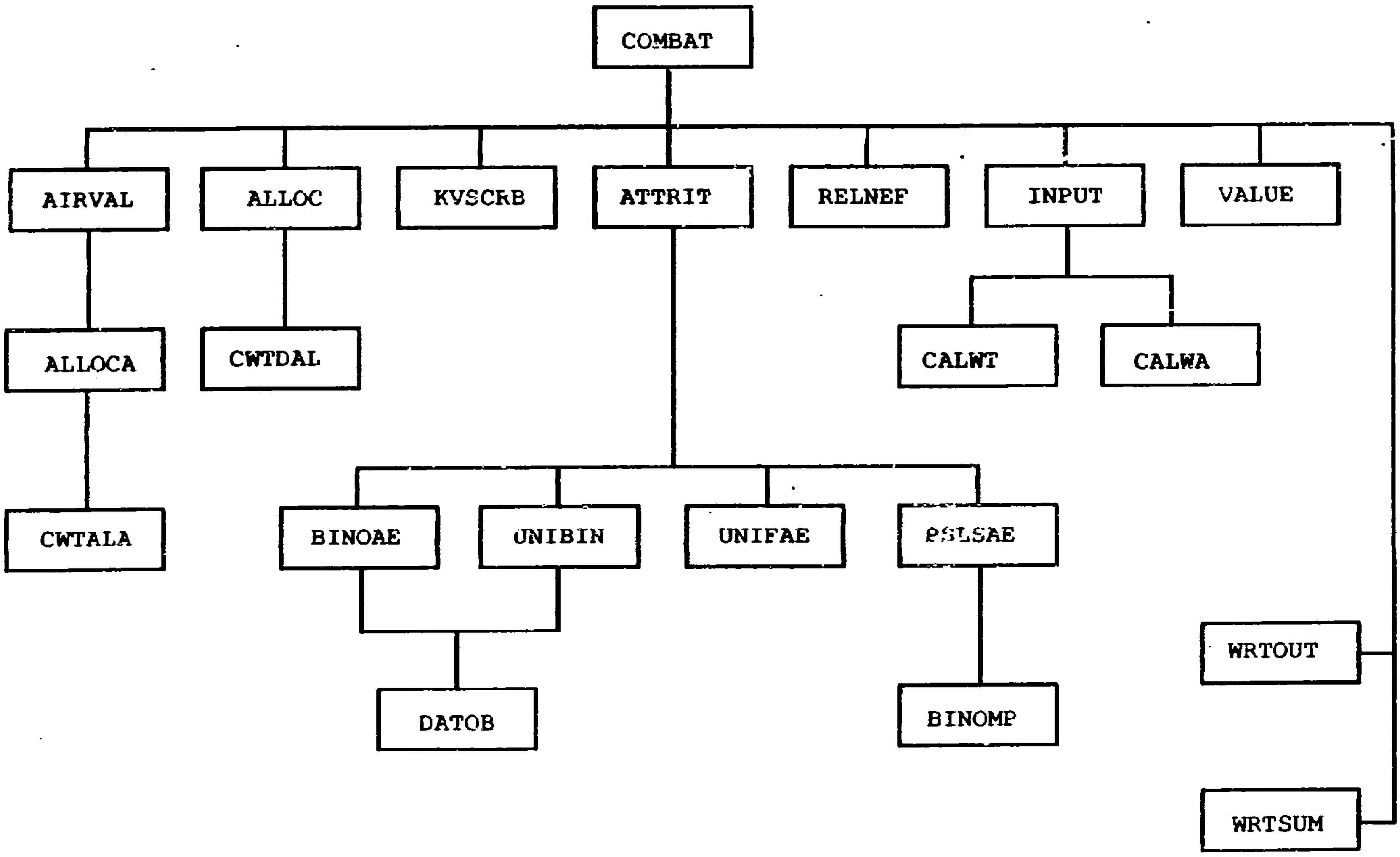


Figure VII-3. A Calling Tree of Subroutines and Functions for Program COMBAT

Table VII-11. The Total Size of COMBAT and of Each of Its Program Units

```
The following data concern program units in File COMBAT   .FOR

In Program COMBAT, the number of:
Initial Lines =  288,  Continuation Lines =    9,  Total Statement Lines =  297;
Real Comments =   17,       Blank Comments =   35,    Total Comment Lines =   52.
                           The total number of lines in Program COMBAT =  349.

The following data concern program units in File AIRVAL  .FOR

In Subroutine AIRVAL, the number of:
Initial Lines =   17,  Continuation Lines =    3,  Total Statement Lines =   20;
Real Comments =    3,       Blank Comments =    2,    Total Comment Lines =    5.
                       The total number of lines in Subroutine AIRVAL =   25.

The following data concern program units in File ALLOC   .FOR

In Subroutine ALLOC , the number of:
Initial Lines =  105,  Continuation Lines =    6,  Total Statement Lines =  111;
Real Comments =   10,       Blank Comments =   10,    Total Comment Lines =   20.
                       The total number of lines in Subroutine ALLOC  =  131.

The following data concern program units in File ALLOCA  .FOR

In Subroutine ALLOCA, the number of:
Initial Lines =   93,  Continuation Lines =    6,  Total Statement Lines =   99;
Real Comments =   11,       Blank Comments =    9,    Total Comment Lines =   20.
                       The total number of lines in Subroutine ALLOCA =  119.

The following data concern program units in File ATTRIT  .FOR

In Subroutine ATTRIT, the number of:
Initial Lines =   57,  Continuation Lines =    2,  Total Statement Lines =   59;
Real Comments =    1,       Blank Comments =    3,    Total Comment Lines =    4.
                       The total number of lines in Subroutine ATTRIT =   63.

The following data concern program units in File BINOAE  .FOR

In Subroutine BINOAE, the number of:
Initial Lines =   23,  Continuation Lines =    1,  Total Statement Lines =   24;
Real Comments =    1,       Blank Comments =    4,    Total Comment Lines =    5.
                       The total number of lines in Subroutine BINOAE =   29.

The following data concern program units in File BINOMP  .FOR

In Subroutine BINOMP, the number of:
Initial Lines =   42,  Continuation Lines =    0,  Total Statement Lines =   42;
Real Comments =    8,       Blank Comments =    5,    Total Comment Lines =   13.
                       The total number of lines in Subroutine BINOMP =   55.

The following data concern program units in File CALWT   .FOR

In Subroutine CALWT , the number of:
```

(continued)

Table VII-11. (Continued)

```
Initial Lines =   17,  Continuation Lines =    2,  Total Statement Lines =   19;
Real Comments =    1,       Blank Comments =    3,    Total Comment Lines =    4.
                          The total number of lines in Subroutine CALWT  =   23.


The following data concern program units in File CALWTA  .FOR

In Subroutine CALWTA, the number of:
Initial Lines =   17,  Continuation Lines =    3,  Total Statement Lines =   20;
Real Comments =    1,       Blank Comments =    3,    Total Comment Lines =    4.
                          The total number of lines in Subroutine CALWTA =   24.


The following data concern program units in File CWTALA  .FOR

In Subroutine CWTALA, the number of:
Initial Lines =   13,  Continuation Lines =    1,  Total Statement Lines =   14;
Real Comments =    1,       Blank Comments =    3,    Total Comment Lines =    4.
                          The total number of lines in Subroutine CWTALA =   18.


The following.data concern program units in File CWTDAL  .FOR

In Subroutine CWTDAL, the number of:
Initial Lines =   13,  Continuation Lines =    1,  Total Statement Lines =   14;
Real Comments =    1,       Blank Comments =    3,    Total Comment Lines =    4.
                          The total number of lines in Subroutine CWTDAL =   18.


The following data concern program units in File DATOB   .FOR

In Function DATOB , the number of:
Initial Lines =   10,  Continuation Lines =    0,  Total Statement Lines =   10;
Real Comments =    0,       Blank Comments =    2,    Total Comment Lines =    2.
                            The total number of lines in Function DATOB  =   12.


The following data concern program units in File INPUT   .FOR

In Subroutine INPUT , the number of:
Initial Lines =  301,  Continuation Lines =   45,  Total Statement Lines =  346;
Real Comments =    5,       Blank Comments =   30,    Total Comment Lines =   35.
                          The total number of lines in Subroutine INPUT  =  381.


The following data concern program units in File KVSCRB  .FOR

In Subroutine KVSCRB, the number of:
Initial Lines =   20,  Continuation Lines =    2,  Total Statement Lines =   22;
Real Comments =    1,       Blank Comments =    3,    Total Comment Lines =    4.
                          The total number of lines in Subroutine KVSCRB =   26.


The following data concern program units in File PSLSAE  .FOR

In Subroutine PSLSAE, the number of:
Initial Lines =  113,  Continuation Lines =    3,  Total Statement Lines =  116;
Real Comments =    1,       Blank Comments =   19,    Total Comment Lines =   20.
                          The total number of lines in Subroutine PSLSAE =  136.
```

(continued)

Table VII-11. (Concluded)

```
The following data concern program units in File RELNEF  .FOR

In Subroutine RELNEF, the number of:
Initial Lines =   47,  Continuation Lines =    1,  Total Statement Lines =   48;
Real Comments =   59,        Blank Comments =   23,    Total Comment Lines =   82.
                         The total number of lines in Subroutine RELNEF =  130.

The following data concern program units in File UNIBIN  .FOR

In Subroutine UNIBIN, the number of:
Initial Lines =   57,  Continuation Lines =    1,  Total Statement Lines =   58;
Real Comments =    4,        Blank Comments =    9,    Total Comment Lines =   13.
                         The total number of lines in Subroutine UNIBIN =   71.

The following data concern program units in File UNIFAE  .FOR

In Subroutine UNIFAE, the number of:
Initial Lines =   37,  Continuation Lines =    1,  Total Statement Lines =   38;
Real Comments =    1,        Blank Comments =    4,    Total Comment Lines =    5.
                         The total number of lines in Subroutine UNIFAE =   43.

The following data concern program units in File VALUE   .FOR

In Subroutine VALUE , the number of:
Initial Lines =  194,  Continuation Lines =    2,  Total Statement Lines =  196;
Real Comments =    6,        Blank Comments =   13,    Total Comment Lines =   19.
                         The total number of lines in Subroutine VALUE  =  215.

The following data concern program units in File WRTOUT  .FOR

In Subroutine WRTOUT, the number of:
Initial Lines =  167,  Continuation Lines =   36,  Total Statement Lines =  203;
Real Comments =    6,        Blank Comments =    4,    Total Comment Lines =   10.
                         The total number of lines in Subroutine WRTOUT =  213.

The following data concern program units in File WRTSUM  .FOR

In Subroutine WRTSUM, the number of:
Initial Lines =  163,  Continuation Lines =   25,  Total Statement Lines =  188;
Real Comments =    7,        Blank Comments =   15,    Total Comment Lines =   22.
                         The total number of lines in Subroutine WRTSUM =  210.


In all of the files listed here, the total number of:
Initial Lines = 1794,  Continuation Lines =  150,  Total Statement Lines = 1944;
Real Comments =  145,        Blank Comments =  202,    Total Comment Lines =  347.
                       The total number of lines in all of these files = 2291.
```

E. RUNNING COMBAT

As stated in Chapter I, a copy of the COMBAT computer program on a 5.25-inch disk (PC/MS-DOS format) has been attached to the inside back cover. If this disk is missing, another copy can be obtained from the authors at the Institute for Defense Analyses. This disk contains the same code (i.e., all of the files listed on Table VII-10), an executable file (COMBAT.EXE), some entirely hypothetical data sets, and the outputs produced by running COMBAT with some of these data sets.

Relevant comments concerning copying and running COMBAT are contained in the README file in the root directory of that disk. This file, which should be read before running COMBAT, is transcribed below.

This disk contains the code and some sample test data
for the computer program described in:

IDA Paper P-2248

COMBAT: A COMPUTER PROGRAM TO INVESTIGATE
AIMED FIRE ATTRITION EQUATIONS, ALLOCATIONS
OF FIRE, AND THE CALCULATION OF WEAPONS SCORES

Lowell Bruce Anderson
Frederic A. Miercort

Institute for Defense Analyses
1801 North Beauregard Street
Alexandria, Virginia 22311

September 1989

This disk contains three subdirectories. Subdirectory CODE contains the source code for this computer program. Subdirectory DATA-EXE contains an executable file and some entirely hypothetical test data sets for this program. Two of these data sets require the user to interactively supply additional data values. The other data sets are essentially self-contained, and the output files produced by these other data sets are contained in Subdirectory OUTPUT. Other than the comments below, this disk contains no documentation or help files. Interested parties should consult the reference cited above.

The executable file in Subdirectory DATA-EXE requires a floating-point coprocessor. The code in Subdirectory CODE must be recompiled and relinked in order to run COMBAT without such a coprocessor. All of the source code needed to compile and link COMBAT is contained in Subdirectory CODE. With the exception of the INCLUDE statements, all of this code is standard FORTRAN-77.

If the (external) DOS command XCOPY is available, this disk can be copied as follows. First, create a suitable subdirectory on the target drive, say drive C. Next, change directories so that this new directory is the current directory. Next, insert this disk into a suitable source drive, say drive A. Finally, execute the DOS command:

```
XCOPY A: C: /S
```

If XCOPY is not available, new subdirectories can be created, and this disk can be copied subdirectory-by-subdirectory.

COMBAT expects to find its input file on the current directory. If COMBAT is run with the current directory being Subdirectory DATA-EXE, it will write results to a file (or to two files, if INVAL = 6) on that subdirectory. These results can then be compared, if desired, to the corresponding output file(s) on Subdirectory OUTPUT.

CAUTION: COMBAT opens its output files on the current directory with STATUS = 'UNKNOWN'. Accordingly, if a file already exists on the current directory with the same name as that being given to a COMBAT output file, running COMBAT will overwrite that previously existing file. See Chapter VII of the reference for a discussion of how output files are named by COMBAT.

REFERENCES

1. Anderson, L.B., *Antipotential Potential*, IDA Note N-845, Institute for Defense Analyses, Arlington VA, April 1979.

2. Borwein, J.M., and P.B. Borwein, "The Way of All Means," *American Mathematical Monthly*, Vol. 94, pp. 519-522, June-July 1987.

3. Törnqvist, L., P. Vartia, and Y.O. Vartia, "How Should Relative Changes Be Measured?," *The American Statistician*, Vol. 39, pp. 43-46, February 1985.

4. Anderson, L.B., *Antagonistic Games*, IDA Paper P-1204, Institute for Defense Analyses, Arlington VA, August 1976.

5. Farnsworth, D., and R. Orr, "Gini Means," *American Mathematical Monthly*, Vol. 93, pp. 603-607, October 1986.

6. Karr, A.F., *A Heterogeneous Linear Law Binomial Attrition Process with Priority Allocation of Fire*, Working Paper WP-17 of IDA Project 2371, Institute for Defense Analyses, Alexandria VA, March 1982.

7. Karr, A.F., *Review and Critique of the Vector-2 Combat Model*, IDA Paper P-1315, Institute for Defense Analyses, Arlington VA, December 1977.

8. Anderson, L.B., *Attrition Papers Referenced in IDAGAM I*, IDA Note N-846, Institute for Defense Analyses, Arlington VA, April 1979.

9. Karr, A.F., *Stochastic Attrition Models of Lanchester Type*, IDA Paper P-1030, Institute for Defense Analyses, Arlington VA, June 1974.

10. Karr, A.F., *A Class of Lanchester Attrition Processes*, IDA Paper P-1230, Institute for Defense Analyses, Arlington VA, December 1976.

11. Karr, A.F., *Lanchester Attrition Processes and Theater-Level Combat Models*, IDA Paper P-1528, Institute for Defense Analyses, Arlington VA, September 1981. This paper also appears as a chapter of: Shubik, M., et al., *Mathematics of Conflict*, North-Holland, Amsterdam, The Netherlands, 1983.

12. Schwartz, E.L., *A Short Proof of the Basic Binomial Heterogeneous Linear Attrition Equation, with Indications for Extensions*, Working Paper WP-21 of IDA Project 2371, Institute for Defense Analyses, Alexandria VA, November 1983, revised June 1984.

13. Sternberg, S.R., *Development of Optimal Allocation Strategies in Heterogeneous Lanchester-Type Processes*, unpublished PhD dissertation, University of Michigan, Ann Arbor MI, 1970. This dissertation is identical to: Sternberg, S.R., *Development of Optimal Allocation Strategies in Heterogeneous Lanchester-Type*

Processes, Technical Report SRL 2147 TR 71-1, Systems Research Laboratory, University of Michigan, Ann Arbor MI, June 1971.

14. Anderson, L.B., J.W. Blankenship, and A.F. Karr, *An Attrition Model for Penetration Processes*, IDA Paper P-1148, Institute for Defense Analyses, Arlington VA, March 1976.

15. Schwartz, E.L., *Attrition Processes with Parking Areas*, IDA Paper P-2144, Institute for Defense Analyses, Alexandria VA, September 1988.

16. Anderson, L.B., *Heterogeneous Point Fire and Area Fire Attrition Processes that Explicitly Consider Various Types of Munitions and Levels of Coordination*, IDA Paper P-2249, Institute for Defense Analyses, Alexandria VA, October 1989.

17. Karr, A.F., *On the Lulejian-I Combat Model*, IDA Paper P-1182, Institute for Defense Analyses, Arlington VA, August 1976. A companion paper is: Karr, A.F., *Additional Remarks on Attrition and FEBA Movement Computations in the Lulejian-I Combat Model*, IDA Paper P-1277, Institute for Defense Analyses, Arlington VA, November 1977.

18. Karr, A.F., *On a Class of Binomial Attrition Processes*, IDA Paper P-1031, Institute for Defense Analyses, Arlington VA, June 1974.

19. Karr, A.F., *A Heterogeneous Linear Law Binomial Attrition Equation*, Working Paper WP-19 of IDA Project 2371, Institute for Defense Analyses, Alexandria VA, September 1983.

20. Karr, A.F., *A Review of Seven DOAE Papers on Mathematical Models of Attrition*, IDA Paper P-1249, Institute for Defense Analyses, Arlington VA, November 1977.

21. Karr, A.F., and E.L. Schwartz, *Modeling of Attrition in a Barrier Penetration Process with Evenly Spaced Barrier Elements*, IDA Paper P-1727, Institute for Defense Analyses, Alexandria VA, forthcoming.

22. Schwartz, E.L., *Some Attrition Models in which Engaged Targets Can Return Fire*, Working Paper WP-52 of IDA Project 3666, Institute for Defense Analyses, Alexandria VA, January 1987.

23. Anderson, L.B., *Some Concepts Concerning the Incorporation of Multiple Attrition Assessments and Night Combat into IDAGAM*, Working Paper WP-8 of IDA Project 3609, Institute for Defense Analyses, Arlington VA, October 1980.

24. Karr, A.F., *Some Questions of Approximation Involving Lanchester and Binomial Attrition Processes*, Working Paper WP-10 of IDA Project 2371, Institute for Defense Analyses, Arlington VA, January 1981.

25. Anderson, L.B., *An Initial Postulation of a Relatively General Attrition Process*, Working Paper WP-20 of IDA Project 2371, Institute for Defense Analyses, Alexandria VA, October 1983, Revised May 1984.

26. Andeison, L.B., et al., *The IDA Defense Planning Model (IDAPLAN) Part II: Effectiveness Module Documentation, Volume 3: Conventional Portion of the Theater Land-Air Module*, Vol. 3 of Part II of IDA Report R-352, Institute for Defense Analyses, Alexandria VA, forthcoming.

27. Anderson, L.B., C.M. Hampton, and J.M. Turley, *Modified Antipotential Potential*, Working Paper WP-10 of IDA Project 3609, Institute for Defense Analyses, Arlington VA, November 1980.

28. Anderson, L.B., *A Heterogeneous Shoot-Look-Shoot Attrition Process*, IDA Paper P-2250, Institute for Defense Analyses, Alexandria VA, October 1989.

29. Anderson, L.B., *Generic Formulas for Calculating Kill Rate Matrices*, Working Paper WP-82 of IDA Project 26-01-13, Institute for Defense Analyses, Alexandria VA, October 1988.

APPENDIX A

AN INTRODUCTION TO ANTIPOTENTIAL POTENTIAL WEAPON SCORES

CONTENTS OF APPENDIX A

1. Foreword A-1
2. Notation A-1
3. Basic Assumption A-1
4. Note 1 .... A-2
5. Note 2 .... A-2
6. Derivation A-2
7. Theorem (Frobenius, 1912) A-3
8. Note 3 .... A-3
9. Scaling Assumptions A-3
10. Rationale for this Choice of Scaling Assumptions A-4
11. Relationship to Lanchester Square A-4
12. Some Limitations of the Antipotential Potential Method A-4
13. Uses of the Antipotential Potential Method A-5
14. Relationship to IDAGAM I A-5

References for Appendix A A-7

Annex A to Appendix A. An Example Why Thrall's Method Should Not be Used to Scale Weapon Values for Force Ratios A-A-1

Annex B to Appendix A. A Result on Firepower Models and Weapon Tradeoffs A-B-1

AN INTRODUCTION TO ANTIPOTENTIAL POTENTIAL WEAPON SCORES

1. Foreword

This appendix was extracted from Chapter III of IDA Note N-845, Reference [A.9]. Following the body of this appendix, two annexes are given that amplify some points made below. Annex A gives an example why Thrall's method should not be used to scale weapon values. Annex B gives a result on firepower models and weapon tradeoffs.

2. Notation

B_i = (input) number of Blue weapons of type i.

R_j = (input) number of Red weapons of type j.

K_{ij}^b = (input) rate at which each Blue weapon of type i is killing Red weapons of type j. Note: for Lanchester square K_{ij}^b = constant,

for IDAGAM I $K_{ij}^b = K_{ij}^b \left(R_1, R_2, \ldots\right)$,

in general $K_{ij}^b = K_{ij}^b \left(B_1, B_2, \ldots, R_1, R_2, \ldots\right)$.

K_{ji}^r = (input) rate at which each Red weapon of type j is killing Blue weapons of type i.

V_i^b = "value" of each Blue weapon of type i (to be computed).

V_j^r = "value" of each Red weapon of type j (to be computed).

β^b = proportionality constant for Blue killing Red (to be computed).

β^r = proportionality constant for Red killing Blue (to be computed).

3. Basic Assumption

The "value" of each Blue weapon of type i is proportional to the sum over j of the rate at which that Blue weapon is killing Red weapons of type j times the "value" of Red weapons of type j.

That is, $V_i^b \propto \sum_j K_{ij}^b V_j^r$ or $\beta^b V_i^b = \sum_j K_{ij}^b V_j^r$.

Similarly, $V_j^r \propto \sum_i K_{ji}^r V_i^b$ or $\beta^r V_j^r = \sum_i K_{ji}^r V_i^b$.

4. Note 1

Since "value" is a function of K_{ij}^b and $K_{ji'}^r$, and since K_{ij}^b can be a function of $R_1, R_2, \ldots$, and K_{ji}^r can be a function of $B_1, B_2, \ldots$, "value" can be a function of the number of weapons on each side at a particular point in time. If we assume a typical (or standard) Blue force and Red force, one can compute typical (or standard) values for Blue and Red.

5. Note 2

$$\beta^b = \frac{\sum_j K_{1j}^b V_j^r}{V_1^b} = \frac{\sum_j K_{2j}^b V_j^r}{V_2^b} = \ldots = \frac{\sum_i B_i \sum_j K_{ij}^b V_j^r}{\sum_{i'} B_{i'} V_{i'}^b}$$

Thus, as K_{ij}^b is the rate at which one Blue weapon of type i is killing Red weapons of type j, β^b is the rate at which one unit's-worth-of-value of Blue weapons is killing Red value. (K_{ij}^b depends on i and j, β^b depends on neither i nor j.)

Similarly, β^r is the rate at which one unit's-worth-of-value of Red weapons is killing Blue value.

6. Derivation

$$\beta^b V_i^b = \sum_j K_{ij}^b V_j^r = \sum_j K_{ij}^b \sum_{i'} K_{ji'}^r V_{i'}^b / \beta^r .$$

Let $\lambda = \beta^b \beta^r$ and $\overline{K}_{ii'} = \sum_j K_{ij}^b K_{ji'}^r$ (i.e., $\overline{K} = K^b K^r$) .

Then $\lambda V_i^b = \sum_{i'} \overline{K}_{ii'} V_{i'}^b$.

λ is called a characteristic value or eigenvalue of the matrix $\overline{K}$, and V^b is called a characteristic vector or eigenvector of the matrix $\overline{K}$ corresponding to λ.

7. **Theorem (Frobenius, 1912)**

If $\overline{K}$ is "irreducible" and non-negative, then there is at least one solution for λ that is strictly positive; and corresponding to the largest such solution there is a solution for V^b that is the unique (up to a scaling constant) non-negative eigenvector of the matrix $\overline{K}$.

8. **Note 3**

λ, V^b, and V^r can be computed recursively (if $\overline{K}$ is not periodic).

Pick any vector $V^b(0)$ (any non-negative, non-zero vector of proper size).

Let $V_j^r(0) = \sum_i K_{ji}^r V_i^b(0)$.

Let $V_i^b(1) = \sum_j K_{ij}^b V_j^r(0)$ and renormalize.

Let $V_j^r(1) = \sum_i K_{ji}^r V_i^b(1)$.

Let $V_i^b(2) = \sum_j K_{ij}^b V_j^r(1)$ and renormalize, and so on.

Then $\lim_{t\to\infty} V_i^b(t) = V_i^b$.

9. **Scaling Assumptions**

Frobenius' Theorem gives λ, and gives V^b and V^r up to scaling constants. Additional assumptions are needed to scale V^b and V^r, and to find β^b and β^r (we know the product, λ, but not the factors). It turns out that two such assumptions are required. Some pairs that have been suggested are:

(i)	$\sum_j V_j^r = 1$ and $\sum_i V_i^b = 1$	(Dare and James [A.2]),
(ii)	$\beta^b = \sum_j V_j^r$ and $\beta^r = \sum_i V_i^b$	(Howes and Thrall [A.3]),
(iii)	$\beta^b = \sum_j V_j^r R_j$ and $\beta^r = \sum_i V_i^b B_i$	(Spudich in TATAWS III [A.1]),
(iv)	$\beta^b = \beta^r$ and $V_{i_0}^b = 1$	(Holter in COMCAP II [A.6]).

Antipotential Potential uses (iv).

10. Rationale for this Choice of Scaling Assumptions

(1) Since $\beta^b = \beta^r$, this choice allows Blue and Red weapons to be compared to each other, not just among themselves.

(2) It turns out that the quantity $\dfrac{\sqrt{\beta^r}\,\sum_j V_j^r R_j}{\sqrt{\beta^b}\,\sum_i V_i^b B_i}$ is constant for any choice of scaling assumptions. With choice (iv), the force ratio $\dfrac{\sum_j V_j^r R_j}{\sum_i V_i^b B_i}$ equals this constant.

(3) This choice is supported by the Lanchester square arguments given in Holter (pp. 248 to 252 of COMCAP II, Reference [A.6]).

(4) This choice allows subdivision (or aggregation) of essentially identical weapons. (See Annex A, below.)

11. Relationship to Lanchester Square

(1) Dare and James [A.2] have proven the following result.

Let $U^b(t) = \sum_i V_i^b B_i(t)$ and $U^r(t) = \sum_j V_j^r R_j(t)$.

If $\dot{R}_j(t) = -\sum_i B_i(t) K_{ij}^b$ and $\dot{B}_i(t) = -\sum_j R_j(t) K_{ji}^r$,

and K^b and K^r are irreducible, then

(i) $\dot{U}^r(t) = -\beta^b U^b(t)$ and $\dot{U}^b(t) = -\beta^r U^r(t)$ implies that

(ii) $\beta^b V_i^b = \sum_j K_{ij}^b V_j^r$ and $\beta^r V_j^r = \sum_i K_{ji}^r V_i^b$,

and (ii) implies (i).

(2) See the references for other notes.

12. Some Limitations of the Antipotential Potential Method

1. "Value" should really be called "lethality potential" because if

$$K^b_{ij} = K^b_{i'j} \text{ for all } j$$

then

$$V^b_i = V^b_{i'}$$

no matter what K^r_{ji} and $K^r_{ji'}$ are for any j.

2. No linear weighting scheme should be used to make weapon trade-offs. (See Reference [A.8] and see Annex B, below.)

13. Uses of the Antipotential Potential Method

1. As part of the computations in a dynamic model (such as in IDAGAM I, see Section 14 below).

2. As a measure of effectiveness based on the output of a dynamic model. An example of such a measure is

$$\frac{\sum_{d=1}^{D} \dot{U}^r(d)/\sum_{d=1}^{D} U^r(d)}{\sum_{d=1}^{D} \dot{U}^b(d)/\sum_{d=1}^{D} U^b(d)}$$

where $\dot{U}^b(d) = \sum_i V^b_i(d)\dot{B}_i(d)$ = the total Blue value lost on day d of a war of length D--$U^b(d)$, $\dot{U}^r(d)$, and $U^r(d)$ are defined similarly--and where $\dot{B}_i(d)$ and $\dot{R}_j(d)$ are computed by the dynamic model.

14. Relationship to IDAGAM I[a]

"Values" are used to make force ratios and force ratios are used only:

(1) for determining FLOT/FEBA movement,

[a] References for IDAGAM I are:

[1] Lowell Bruce Anderson, Jerome Bracken, James G. Healy, Mary J. Hutzler, and Edward P. Kerlin, *IDA Ground-Air Model I (IDAGAM I), Vol. 1: Comprehensive Description; Vol. 2: Definitions of Variables; Vol. 3: Detailed Description of Selected Aspects; Vol. 4: Computer Program Documentation; Vol. 5: Testing* , IDA Report R-199, Institute for Defense Analyses, Arlington, VA, October 1974.

[2] Lowell Bruce Anderson, Debbie Bennett, and Mary J. Hutzler, *Modifications to IDAGAM I*, IDA Paper P-1034, Institute for Defense Analyses, Arlington, VA, October 1974.

(2) for making reinforcement and withdrawal decisions, and

(3) (optionally) for determining the intensity of losses (i.e., a scale factor).

In particular these values are not used to compute the potential numbers of weapons lost on each side $\left(\dot{B}_i^p \text{ and } \dot{R}_j^p\right)$; they (optionally) can be used to scale these potential losses to determine actual losses $\left(\dot{B}_i \text{ and } \dot{R}_j\right)$. That is, these values are used to compute force ratios, which (optionally) can be used to compute intensity scaling factors α^b and α^r, and then $\dot{B}_i$ and $\dot{R}_j$ can be computed by setting $\dot{B}_i = \alpha^b \dot{B}_i^p$ and $\dot{R}_j = \alpha^r \dot{R}_j^p$.

REFERENCES FOR APPENDIX A

1. ORIGINAL SOURCES

A.1. Spudich, John, *The Relative Kill Productivity Exchange Ratio Technique*, Booz-Allen Applied Research, Inc., Combined Arms Research Office, n.d. (This reference is virtually identical to U.S. Army, Combat Developments Command, Headquarters, TAB E, Appendix II to Annex L, *Tank, Anti-Tank and Assault Weapons Requirements Study*, Phase III (TATAWS III), December 1968.

A.2. Dare, D.P., and B.A.P. James, *The Derivation of Some Parameters for a Corps/Division Model from a Battle Group Model*, Defence Operation Analysis Establishment Memorandum 7120, Ministry of Defence, West Byfleet, United Kingdom, July 1971.

A.3. Thrall, Robert M., and Associates, Chapter 2.C, *Final Report to U.S. Army Strategy and Tactics Analysis Group* (RMT-200-R4-33), May 1972. (This reference is virtually identical to David R. Howes and Robert M. Thrall, "A Theory of Ideal Linear Weights for Heterogeneous Combat Forces," *Naval Research Logistics Quarterly*, Vol. 20, No. 4, December 1973.)

A.4. Johnsrud, Alan E. (Office of the Coordinator of Army Studies), *Firepower Comparison of Weapons and Units*, unpublished paper presented at a conference held at the General Research Corporation, McLean VA, on June 14 and 15, 1973.

2. OTHER SOURCES

A.5. Anderson, Lowell Bruce, *A Method for Determining Linear Weighting Values for Individual Weapons Systems*, Working Paper WP-4 of the Improved Methodologies for General Purpose Forces Planning (New Methods) Study, Institute for Defense Analyses, Arlington VA, April 1973.

A.6. Holter, William H., et al., *Appendix D: NATO Combat Capabilities Analysis II (COMCAP II)*, GRC Report OAD-CR-8, General Research Corporation, McLean VA, August 1973.

A.7. Anderson, Lowell Bruce, *References on Anti-Potential Potential (The Eigenvector Method for Computing Weapon Values)*, Working Paper WP-2 of IDA Project 23-31, Institute for Defense Analyses, Arlington VA, March 1974.

A.8. Anderson, Lowell Bruce, *Some Experiments on the Anti-Potential Potential Method for Computing Weapon Values*, Working Paper WP-5 of IDA Project 23-31, Institute for Defense Analyses, Arlington VA, July 1974.

A.9. Anderson, Lowell Bruce, *Antipotential Potential*, IDA Note N-845, Institute for Defense Analyses, Arlington VA, April 1979.

ANNEX A TO APPENDIX A

AN EXAMPLE WHY THRALL'S METHOD SHOULD NOT BE USED TO SCALE WEAPON VALUES FOR FORCE RATIOS

As in the text above, let

$B = (B_i)$ = the number of Blue weapons (of type i),

$R = (R_j)$ = the number of Red weapons (of type j),

$K^b = \left[K^b_{ij}\right]$ = the Blue kills Red rate matrix,

$K^r = \left[K^r_{ji}\right]$ = the Red kills Blue rate matrix,

$V^b = \left(V^b_i\right)$ = the Blue "value" vector to be computed,

$V^r = \left(V^r_j\right)$ = the Red "value" vector to be computed,

β^b = the Blue proportionality constant,

β^r = the Red proportionality constant,

so that $\beta^b V^b = K^b V^r$ and $\beta^r V^r = K^r V^b$ according to the eigenvector approach for computing weapon values. Also as discussed in the text above, two scaling assumptions are needed to compute specific values for V^b and V^r. Thrall's method uses the scaling assumptions that:

$$\beta^b = \sum_j V^r_j \text{ and } \beta^r = \sum_j V^b_i \, ,$$

whereas Holter's method assumes:

$$\beta^b = \beta^r \text{ and } V^b_1 = 1 \, .$$

Suppose that there is one type of weapon on each side and $K^b = K^r = 10$. Then

$$\beta^h = \beta^r = 10$$

and

$$V^b = V^r = \begin{cases} 10 \text{ by Thrall's method} \\ 1 \text{ by Holter's method.} \end{cases}$$

Thus, if B = R = 100, the force ratio according to either method is:

$$\frac{\left(V^b\right)^T B}{\left(V^r\right)^T R} = 1 .$$

Now suppose that Blue has two "very similar" types of weapons and that B = $\left(B_1, B_2\right) = (50,50)$. Then it is reasonable to assume that

$$K^b = \begin{bmatrix} 10 \\ 10 \end{bmatrix} \text{ and } K^r = [5,5] \text{ (not } K^r = [10,10]).$$

Then, using Thrall's method, $V_1^b = V_2^b = 10$, $V^r = 5$, $\beta^b = 5$, and $\beta^r = 20$.

Using Holter's method, $V_1^b = V_2^b = 1$, $V^r = 1$, $\beta^r = 10$, and $\beta^r = 10$.

Thus, the force ratio according to Thrall's method becomes

$$\frac{\left(V^b\right)^T B}{\left(V^r\right)^T R} = \frac{(10 \times 50) + (10 \times 50)}{5 \times 100} = 2,$$

whereas the force ratio according to Holter's method remains at 1 (as it logically should).

ANNEX B TO APPENDIX A

A RESULT ON FIREPOWER MODELS AND WEAPON TRADEOFFS[1]

It has been proposed that if weapon tradeoffs are made using a "pure" firepower model, or any other "linear model," then the weapon type with the most firepower per unit cost would always be selected. Below is one way to rigorously state and prove that proposal.

Let

M = the number of different types of weapons,

W_i = the number of weapons of type i ($1 \le i \le M$),

V_i = the positive (linear) value of each weapon of type i ($1 \le i \le M$),

C_i = the positive (linear) cost of each weapon of type i ($1 \le i \le M$),

Y = the total firepower of the force = $\sum_{i=1}^{M} W_i V_i$,

K = an arbitrary set,

$f_k(Y)$ = a non-decreasing function of Y for each $k \varepsilon K$,

e.g., $f_1(Y)$ = (casualties to the force in question)$^{-1}$,

$f_2(Y)$ = casualties to opponent,

$f_3(Y)$ = FEBA movement rate, and

let j be such that $\frac{V_j}{C_j} \ge \frac{V_i}{C_i}$ for all i = 1,...,M.

THEOREM: For all $B \ge 0$,

$$\max_{W_1,\dots,W_M} f_k\left(W_1V_1 + \dots + W_MV_M\right)$$

such that

[1] This annex is essentially a reprint of: Anderson, L.B., *A Result on Firepower Models and Weapon Tradeoffs*, Working Paper WP-5 of IDA Project 2344, Institute for Defense Analyses, Arlington VA, October 1975.

$$C_1 W_1 + \ldots + C_M W_M \leq B$$

$$W_i \geq 0 \text{ for all } i \; (1 \leq i \leq M)$$

occurs at

$$W_i^* = \begin{cases} B/C_j & i = j \\ 0 & i \neq j \end{cases}$$

for all k ε K.

NOTE: The point at which this maximum occurs is not necessarily unique.

PROOF: Clearly $\left(W_1^*,\ldots,W_M^*\right)$ satisfies the constraints. Now let $\left(W_1,\ldots,W_M\right)$ be any set of weapons that satisfies the constraints. Then

$$\sum_i W_i V_i = \sum_i \left(\frac{V_i}{C_i}\right) C_i W_i \leq \sum_i \left(\frac{V_j}{C_j}\right) C_i W_i \leq \frac{B}{C_j} V_j = \sum_i W_i^* V_i .$$

Thus

$$f_k(W_1 V_1 + \ldots + W_M V_M) \leq f_k\,(W_1^* V_1 + \ldots + W_M^* V_M)$$

for all k ε K since $f_k(\cdot)$ is non-decreasing for all k ε K.

APPENDIX B

A ROBUST METHOD FOR CONSIDERING EFFECTIVENESS PARAMETERS THAT ARE FUNCTIONS OF ATTACK AND DEFENSE

Many models of combat between two sides (say Red and Blue) allow selected subsets of the input parameters to depend on whether Blue is attacking Red or Red is attacking Blue. For example, there could be separate input engagement rates, allocations of fire, and/or probabilities of kill for Blue on attack against Red, Blue on defense against Red, Red on attack against Blue, and Red on defense against Blue.

When considering such separate-for-attack-and-defense parameters, there are four possible cases: 1) Red on attack and Blue on defense, 2) Blue on attack and Red on defense, 3) both sides attempting to attack, and 4) neither side attempting to attack. Many of the models that allow selected input parameters to depend on whether a side is attempting to attack also require the assumption that exactly one of the cases out of these four possible cases is occurring throughout the entire region of interest over the entire time interval in question. (These models may also assume that this one case must either be Blue attacking and Red defending, or be Red attacking and Blue defending, instead of allowing any one of the four cases to apply.)

Whether such an "exactly one case is occurring" assumption is appropriate is debatable. For example, in addition to being potentially inherently unrealistic, such an assumption can cause significant anomalies to occur--such as a small change in one input causing a force ratio to go slightly over a threshold, which causes a different side to attack, which then yields significantly different results throughout the rest of the war being simulated.

Fortunately, in this case, it is easy to avoid these anomalies. That is, there is no good reason to make this always-attack or always-defend assumption. A model can just as easily be constructed to allow weighted averages of attack and defense values to be used. Such a model would be robust concerning the type of anomaly just described in that very small changes in the inputs would not produce relatively large changes in results due to a different side going on the attack.

For example, to develop such a model let f^b denote the portion of the region in question that Blue will be on the attack times the portion of the time interval in question that Blue would be attacking in this portion of this region, and let f^r be defined analogously for Red. Thus, $0 \leq f^i \leq 1$ for $i \in \{b,r\}$. The fractions f^b and f^r are not (necessarily) direct inputs--generally they would be determined as functions of force ratio (and, perhaps, of other arguments). Let x^b denote the relevant results of combat using parameters appropriate for the case in which Blue is on the attack and Red is on the defense; and let x^r

denote these results for Red on the attack and Blue on the defense. Let x^m denote these results for a "meeting engagement" case, i.e., the results using parameters appropriate for the case in which both sides are attempting to attack, and let x^h denote these results for a "holding" case, i.e., the results using parameters appropriate for the case in which both sides are on defense. It is assumed that the model can use the appropriate parameter sets to calculate x^b, x^r, x^m, and x^h. Let y denote the overall results to be calculated from the x's and the f's. Then, y can be calculated as

$$y = \begin{cases} f^b x^b + f^r x^r + (1-f^b-f^r)x^h & f^b + f^r < 1 \\ f^b x^b + f^r x^r & f^b + f^r = 1 \\ (1-f^r)x^b + (1-f^b)x^r + (f^b+f^r-1)x^m & f^b + f^r > 1\,. \end{cases}$$

In particular, note that the functions that yield f^b and f^r need not necessarily satisfy the property that $f^b + f^r = 1$. These functions need only satisfy the general consistency property that $0 \leq f^i \leq 1$ for $i \in \{b,r\}$.

APPENDIX C

AN IMPROVED METHOD FOR DEGRADING THE NOMINAL EFFECTIVENESS OF UNBALANCED FORCES

It has been argued that, in modeling combat, the nominal effectiveness of a force should be degraded if that force is significantly unbalanced. For example, a force consisting only of artillery and aircraft could be considered as being unbalanced and hence as not being an effective force because it could easily be overrun by enemy armor and infantry. Conversely, a more balanced force consisting of fewer artillery and aircraft (but with some armor and infantry) could be quite effective because the armor and infantry, in addition to providing their own firepower, help protect artillery and air bases.

The approach used in IDAGAM (Reference [C.1]) to address the degradation of unbalanced forces is to assume that each type of ground weapon can be put into one of three classes. A type of ground weapon is in class one if it can operate on the battlefield without any other weapons around it for self-protection. A weapon type is in class two if it needs weapons in class one or other adequately protected weapons in class two to protect it, and once adequately protected, it can protect other weapons in class two as well as weapons in class three. A weapon type is in class three if it needs weapons in class one or protected weapons in class two to protect it, and it cannot protect other weapons. For example, infantry could be in class one, tanks in class two, and artillery in class three.

While this approach seems generally reasonable, it was implemented in a flawed manner in IDAGAM. The first flaw is that it only considers ground weapons, not combat aircraft. The second flaw is as follows. IDAGAM assumes that each weapon in class one can simultaneously protect an input number of weapons of each type in each of the other two classes. A more reasonable assumption would be that each weapon in class one can protect an input number of notional (or typical) weapons (in total) from the other two classes. This protection would be prorated across the types of weapons in both other classes in proportion to the (weighted) number of weapons in these classes needing protection, which would determine the number of protected weapons of each type in each class. The analogous comment applies to the consideration of the protective capabilities of weapons in class two.

One way to implement this (more reasonable) assumption is as follows. Let

N = the number of different types of weapons systems (including aircraft) being simulated in the force in question.

For $i=1,\ldots,N$ let

n_i = the number of weapons systems of type i on (or capable of flying over) the battlefield that have ammunition and are ready to fight (but may or may not need protection provided by other weapons in order to participate in combat) for the force in question, and

$$a_i = \begin{cases} 1 & \text{if weapons of type i are in class one (i.e., they need no protection from other weapons in order to participate in combat)} \\ 0 & \text{if weapons of type i are in class two or class three .} \end{cases}$$

Let c_i be a weighting factor such that each weapon of type i in class two or three corresponds to c_i (strictly positive) notional weapons for the purpose of prorating protection across the types of weapons that need to be protected, and let c_i be zero if weapons of type i are in class one. (For example, a tank might correspond to two notional weapons, so $c_i = 2$ when i denotes tanks; while a mortar might correspond to half of a notional weapon, so $c_i = 0.5$ when i denotes mortars.) Thus

$$c_i \in \begin{cases} \{0\} & \text{if weapons of type i are in class one} \\ (0,\infty) & \text{if weapons of type i are in class two or three .} \end{cases}$$

When i denotes a type of weapon in class one, let ρ_i be the total number of notional weapons in classes two and three that each weapon of type i can protect. When i denotes a type of weapon in class two, let ρ_i be the total number of notional weapons in classes two and three that $1/c_i$ protected weapons of type i can protect. (Note that $1/c_i$ weapons of type i is one notional weapons' worth of weapons of type i.) When i denotes a type of weapon in class three, let ρ_i be zero. Since protected weapons in class two can protect other weapons in class two, assume that ρ_i is strictly less than one when i denotes a type of weapon in class two. Thus

$$\rho_i \in \begin{cases} [0,\infty) & \text{if i denotes weapons in class one} \\ [0,1) & \text{if i denotes weapons in class two} \\ \{0\} & \text{if i denotes weapons in class three .} \end{cases}$$

The quantities N, n_i, a_i, c_i, and ρ_i are inputs to this method. The output of this method is the number of weapons of type i, denoted below by m_i, that can participate in combat. To calculate m_i from these inputs, let

$$x_i = \begin{cases} c_in_i / \sum_{j=1}^{N} c_jn_j & c_in_i > 0, \\ 0 & c_in_i = 0\,. \end{cases}$$

Note that x_i is zero for all i that denote weapons in class one, $x_i \in [0,1]$ for all i that denote weapons in class two or three, and $\sum_i x_i = 1$ if $c_jn_j > 0$ for any j. The term x_i can be interpreted as the fraction of weapons needing protection that are of type i out of all of the weapons that need protection, measured in terms of notional weapons. Let

$$q = \sum_{i=1}^{N} \rho_i x_i\,.$$

Note that $x_i\rho_i = 0$ unless i denotes a weapon type in class two, and so $0 \le q < 1$ since $0 \le \rho_i < 1$ for all i that denote weapons in class two, Now let

$$y_i = \begin{cases} 0 & \text{if i denotes a type of weapon in class one} \\ x_i/c_i & \text{otherwise}\,, \end{cases}$$

and let

$$z = \sum_{i=1}^{N} \rho_i a_i n_i\,,$$

so that z is the total number of notional weapons that can be protected by all weapons in class one.

Given that z notional weapons can be protected by the weapons in class one, the particular number of weapons of type i that can be protected by the weapons in class one is y_iz. For i denoting weapons in class two, these y_iz protected weapons can protect additional weapons in classes two and three, the ones of which that are in class two being able to protect still more weapons and so on. Accordingly, the total number of weapons of type i that can be protected is given by

$$y_i z + y_i(\Sigma_j \rho_j c_j y_j z) + y_i(\Sigma_j \rho_j c_j y_j(\Sigma_k \rho_k c_k y_k z)) + \ldots$$

$$= y_i z + y_i q z + y_i q^2 z + \ldots$$

$$= y_i z/(1-q) \ .$$

Thus

$$m_i = \begin{cases} n_i & \text{if i denotes a type of weapon in class one} \\ \min\{n_i, y_i z/(1-q)\} & \text{if i denotes a type of weapon in class two or three} \end{cases}$$

where, as defined above, m_i gives the number of ready weapons of type i that either need no protection or are sufficiently protected by other weapons so that they can participate in combat during the time period in question.

Unprotected weapons (necessarily of classes two or three) would not participate in combat and so would not be vulnerable to enemy fire and, more importantly, could not fire at enemy targets. As page 52 of Volume 3 of Reference [C.1] states:

> The point of these calculations is not that weapons are either fully protected and fully effective or not protected and withdrawn from battle; the point is that the extremes where a force becomes too unbalanced can be roughly estimated--and that the model should have some way of incorporating these rough estimates, so as not to assign full effectiveness to an unbalanced force.

REFERENCE FOR APPENDIX C

C.1. Anderson, L.B., J. Bracken, J.G. Healy, M.J. Hutzler, and E.P. Kerlin, *IDA Ground-Air Model I (IDAGAM I)*, Volume 1: *Comprehensive Description*, Volume 2: *Definitions of Variables*, Volume 3: *Detailed Descriptions of Selected Portions*, Volume 4: *Documentation*, and Volume 5: *Testing*, IDA Report R-199, Institute for Defense Analyses, Arlington VA, October 1974.

APPENDIX D

A GENERAL METHOD FOR RELATING BOUNDS ON ATTRITION TO THE AVERAGE RATE OF MOVEMENT OF GROUND FORCES

CONTENTS OF APPENDIX D

1. Background D-1

2. Three Criteria for Considering Attrition Versus FEBA Movement Tradeoffs....... D-3

3. The Proposed Method for Considering Attrition Versus FEBA Movement Tradeoffs D-4

References for Appendix D D-7

A GENERAL METHOD FOR RELATING BOUNDS ON ATTRITION TO THE AVERAGE RATE OF MOVEMENT OF GROUND FORCES

1. Background

It has long been argued that a force engaged in combat can trade attrition for territory. In particular, it has frequently been argued that a force engaging in "full" combat can lower the rate of attrition it is suffering at the expense of a less favorable (or more unfavorable) movement of the Forward Edge of the Battle Area (FEBA) by participating less fully in that combat. Many (frequently older) models of combat compute both attrition and FEBA movement as functions of force ratios, and these models are generally able to implicitly trade off attrition for FEBA movement via these functions. However, several modern combat models (and some options in some older models) do not compute attrition directly as a function of force ratio, so if these models are to consider attrition versus FEBA movement tradeoffs, these tradeoffs must be considered explicitly.

Explicitly incorporating attrition versus FEBA movement tradeoffs has turned out to be, in some sense, harder than it might first appear. Indeed, the majority of large-scale combat models do not explicitly incorporate such tradeoffs. Two models that attempt to incorporate these tradeoffs explicitly are a model developed by Lulejian and Associates (Reference [D.1]) in 1974 and a model developed by Joshua Epstein (Reference [D.2]) in 1985. However, both of these models incorporate attrition versus FEBA movement tradeoffs in a relatively complex and flawed manner. The flaws of Reference [D.1] are described in References [D.3] and [D.4]. Some of the flaws of Reference [D.2] are as follows. (It should be noted that a major strength of Reference [D.2] is that it points out the aforementioned potential defect concerning attrition versus FEBA movement tradeoffs in modern combat models.)

First, Reference [D.2] contains an extremely simplistic model of ground-to-ground combat (e.g., it is homogeneous in weapon types), and no indication is given as to how to extend the model to be more realistic.† Thus, Reference [D.2] offers no help to analysts who want to address attrition versus FEBA movement tradeoffs, but who wish to do so

† The model in Reference [D.2] also contains an extremely simplistic representation of air interactions, but it is clear how, in general, to extend the air portion of that model to more realistic representations.

using a model with more features (e.g., with adequate representation of various types of ground weapons) than are found in Reference [D.2].‡

Second, Reference [D.2] requires that one side always be on the attack and the other side always be on the defense throughout the entire theater and throughout the entire war. In reality, each side might attack (or counterattack) in various parts of the theater at various times in the war, and might be on defense in other parts of the theater and/or at other times in the war. (See Appendix B above for further discussion of this aspect of combat.) The "only one side can attack" assumption in Reference [D.2] is quite significant concering the methodology Reference [D.2] proposes because: (1) the methodology appears to be very sensitive as to which side is labeled as the attacker, and (2) it treats the side labeled as the attacker very differently from the way that it treats the side labeled as the defender. For example, analyses using the model in [D.2] of two forces that are about equal in all respects could produce very different results depending on which side was declared to be the attacker.

Third, the attrition structure in [D.2] contains several logical conundrums. For example, the attrition to the attacker on day one depends on the input $\alpha_g(1)$ but not on the size of the defender's force. If the attacker can choose $\alpha_g(1)$, why not choose $\alpha_g(1) = 0$? If $\alpha_g(1)$ is an independent input, then how is it that the defender can always kill $\alpha_g(1)A_g(1)$ attackers on day one no matter how big $A_g(1)$ is and no matter how small $D_g(1)$ is, where $A_g(1)$ and $D_g(1)$ denote the sizes of the attacker and defender forces, respectively, at the

‡ Combined arms combat, especially such combat at the corps and theater level, inherently involves important interactions among qualitatively and quantitatively different types of air and ground weapons. Of course, a model of as complex a process as corps or theater level combat need not explicitly simulate every aspect of every resource and interaction to be useful. Judgment is needed to determine which aspects of complex processes should be explicitly simulated within computer models and which aspects are better reflected implicitly in the inputs. Some relevant questions here are: Do heterogeneous interactions occur sufficiently often to be important, are such interactions relatively significant, and does the combat modeling community know how to model these aspects sufficiently well that automation is practical and would be a significant improvement over implicit consideration? The answers to all these questions seem to be clearly and definitively positive. Heterogeneous interactions occur continually in combat. Sometimes these heterogeneous interactions can be adequately represented by homogeneous models, frequently they cannot, and often it is not clear until after the fact whether a homogeneous representation was adequate for a particular battle or whether heterogeneous considerations should have been employed. Accordingly, in terms of frequency, significance, and the difficulties involved in using implicit representations, it is quite desirable to simulate (what is necessarily heterogeneous) combat using a heterogeneous model. Heterogeneous Lanchester attrition equations and heterogeneous combat models below theater level have been available since well before 1970. In addition to Chapter V above, see Reference [D.5] for a discussion of heterogeneous attrition equations and their use in theater level models through 1981.

start of day one? The calculation of attrition to the attacker on succeeding days raises similar but more complex questions. Also, the attrition structure of [D.2] is based on an input parameter (denoted by ρ) which is defined to be the attacker's ground lethality killed per defender's ground lethality killed. Reference [D.2] states that "In this model ρ is interpreted as a constant, an average." That ρ is an average over time and space is, as argued in [D.2], quite reasonable. That ρ is independent of both the size of attacking force and of the size of the defending force is not reasonable, and no justification for this latter type of independence is given in [D.2].

The basic problem with References [D.1] and [D.2] may be that, rather than just incorporating attrition versus FEBA movement tradeoffs in a simple and straightforward manner, they introduce too much complexity as part of these tradeoffs. Some complexity is necessary (as described below), but perhaps not as much as introduced in [D.1] and [D.2].

2. Three Criteria for Considering Attrition Versus FEBA Movement Tradeoffs

First, it would be desirable to incorporate attrition versus FEBA movement tradeoffs into a model in such a manner that the outputs of the model are continuous functions of the inputs. That is, one would like to avoid cases in which very small changes in inputs lead to very large changes in outputs. This problem can easily occur, for example, if the outputs are discontinuous functions of which side is on the attack (as in [D.2]) and if the model assumes that exactly one of the two sides must be on the attack throughout the entire area and time period of interest (also as in [D.2]). This problem can also readily occur, for (another) example, if the outputs are discontinuous functions of input "threshold" parameters (again as in [D.2]).

Second, it would be desirable to address the following question in a straightforward yet continuous (as described above) way. Suppose that both Blue and Red postulate a maximum attrition rate beyond which each will trade FEBA movement for attrition to keep within their maximum attrition rate. Suppose also, that, if neither side were to trade territory for attrition, then both sides would exceed their maximum attrition rate. What happens to the attrition and FEBA movement?

Third, it would be desirable to have a method that could incorporate attrition versus FEBA movement tradeoffs into existing models in such a way that, if the attrition to each side turned out to be sufficiently small (e.g., below an input maximal attrition rate for each side), then the attrition and FEBA movement results of those models would be unchanged.

The following approach satisfies these three criteria, and so could easily be incorporated into existing models.

3. The Proposed Method for Considering Attrition Versus FEBA Movement Tradeoffs

Consider the following notation:

N_s = the number of types of weapons on side s, $s=1,2$.

I_{si} = the number (inventory) of weapons of type i on side s in combat at the start of the time period in combat in the region in question, $i=1,\ldots,N_s$, $s=1,2$.

$\hat{I}_{si}$ = the number of weapons of type i on side s that would be killed by enemy fire during the time period in the region in question if both sides fought with full effort throughout the time period, $i=1,\ldots,N_s$, $s=1,2$.

$\hat{E}$ = the change in FEBA position (i.e., the resulting FEBA position minus the FEBA position at the start of the time period) that would occur if both sides fought at full effort throughout the time period in the region in question.

w_{si} = a weighting factor to be applied to weapons of type i on side s in order to compute an aggregated measure of strength, $i=1,\ldots,N_s$, $s=1,2$.

m_s = a maximum loss rate such that, if side s perceives that it would suffer a loss of aggregated strength due to enemy fire at a rate greater than m_s over the region and time period in question if both sides were to fight with full effort throughout that region and time period, then that side will stand back over a portion of the region and/or over a portion of the time period in order to keep its loss rate less than or equal to m_s $(0 < m_s \leq 1)$, $s=1,2$.

$E_s(x)$ = the (algebraic) change in FEBA position that would occur if side s were willing to fight at full effort throughout the time period, but side s' (where $s' = 3-s$) chose to stand back throughout the time period, and the side s over side s' force ratio is given by x, $s=1,2$.

All of these terms would be inputs to this set of computations. The term N_s is a direct input to the model. The term I_{si} could be an input for the first time period (or could be calculated from other inputs for that period) and would be updated to account for attrition and replacements for succeeding time periods. The terms $\hat{I}_{si}$ and $\hat{E}$ would be calculated by the model before attrition versus FEBA movement tradeoffs are considered (and so this approach makes direct use of the existing structure in a model for calculating attrition and FEBA movement). Note that $\hat{E}$ might have been calculated as a function of force ratio, in a manner consistent with the way that $E_s(x)$ is postulated to be a function of force ratio. The tenns w_{si} and m_s could be direct inputs to the model or could be calculated from other inputs. Given x, the term $E_s(x)$ would be calculated from those inputs that specify E_s as a function of x.

Given these (direct or indirect) inputs, the goal of this set of computations is to calculate:

$\acute{I}_{si}$ = the number of weapons of type i on side s that are killed by enemy fire in the time period and region in question considering that one or both sides may not be fighting at full effort (i.e., may be standing back) in portions of the region and/or time period in order to reduce attrition at the possible expense of favorable FEBA movement, and

$\acute{E}$ = the average change in FEBA position that occurs in the region and time period in question considering that one or both sides may not be fighting at full effort in portions of the region and/or time period in order to reduce attrition at the possible expense of favorable FEBA movement.

Formulas that compute $\acute{I}_{si}$ and $\acute{E}$ in a continuous and relatively straightforward manner are as follows. Let

$a_s = \Sigma_i w_{si}\hat{I}_{si} / \Sigma_j w_{sj}I_{sj}$

= the aggregate attrition rate that side s would suffer if both sides were to fight at full effort throughout the time period and region in question, s=1,2,

$p_s = \min\{m_s/a_s, 1\}$

= the portion of the time period and region in question in which side s fights at full effort, s=1,2,

$q_3 = p_1p_2$

= the portion of the time period and region in question in which both sides fight at full effort, and

$q_s = p_s(1-p_{s'})$

= the portion of the time period and region in question in which side s fights at full effort while side s' stands back (s'=3–s), s=1,2.

Then, assuming that attrition due to enemy fire occurs only when neither side is standing back, $\acute{I}_{si}$ can be computed as

$$\acute{I}_{si} = q_3\hat{I}_{si}$$

for $i=1,\ldots,N_s$ and $s=1,2$. Now let

$$x_s = \Sigma_i\, w_{si}(I_{si}-\acute{I}_{si}) \,/\, \Sigma_j\, w_{s'j}(I_{s'j}-\acute{I}_{s'j})$$

for $s'=3-s$, and $s=1,2$, so that x_s is the side s over side s' force ratio after attrition has been assessed. Then $\acute{E}$ can be calculated as

$$\acute{E} = q_3\hat{E} + \Sigma_s\, q_s E_s(x_s).$$

REFERENCES FOR APPENDIX D

D.1. Lulejian and Associates, *The Lulejian-I Theater-Level Model*, WSEG Report 259, Weapons Systems Evaluation Group of the Office of the Director of Defense Research and Engineering, Arlington VA, 1974.

D.2. Epstein, J.M., *The Calculus of Conventional War*, The Brookings Institution, Washington DC, 1985.

D.3. Karr, A.F., *On the Lulejian-I Combat Model*, IDA Paper P-1182, Institute for Defense Analyses, Arlington VA, August 1976.

D.4. Karr, A.F., *Additional Remarks on Attrition and FEBA Movement Computations in the Lulejian-I Combat Model*, IDA Paper P-1277, Institute for Defense Analyses, Arlington VA, November 1977.

D.5. Karr, A.F., *Lanchester Attrition Processes and Theater-Level Combat Models*, IDA Paper P-1528, Institute for Defense Analyses, Arlington VA, September 1981.

APPENDIX E

A GENERALIZED METHOD FOR CALCULATING FORCE STRENGTHS

Given a score, V_i^s, for each resource of type i on side s, Section E.1 of Chapter II defines the force strength for side s as

$$S^s = \sum_{i=1}^{N^s} V_i^s W_i^s ,$$

where W_i^s is the number of resources of type i on side s. An alternative definition would be to define the force strength S_ρ^s as

$$S_\rho^s = \left(\sum_{i=1}^{N^s} V_i^s W_i^s \right)^\rho$$

for any positive value of ρ. With ρ specified, S_ρ^s could be used in place of S^s in all of the measures considered in Section E of Chaper II. Obviously, setting $\rho = 1$ reproduces the measures as defined in Chapter II. The following reasoning suggests alternative values of ρ for PEXPOT, LEVPOT, and DYNPOT.

Consider the following homogeneous case of the heterogeneous structure presented in Chapter II. Suppose that there is one type of weapon on each side (i.e., $N^1 = N^2 = 1$). For simplicity, let the number of weapons on side s be denoted by W_s here instead of by W_1^s as in Chapter II, let the kill rates be denoted by K_s here instead of by K_{11}^s as in Chapter II, and let the score for weapons on side s be denoted by V_s here instead of by $V_1^s(v)$ for $v = 1,...,5$ as in Chapter II. (The choice of methods used to calculate V_s will be clear in the discussion below.)

In terms of this notation, the scores and force ratios produced by the five methods discussed in Chapter II are as follows:

Method	V_1	V_2	$R = W_2V_2/W_1V_1$
APP	1	$(K_2/K_1)^{1/2}$	$W_2(K_2)^{1/2} / W_1(K_1)^{1/2}$ $= (W_2/W_1)(K_2/K_1)^{1/2}$
APPVUL	1	$(K_2/K_1)^{1/2}$	$W_2(K_2)^{1/2} / W_1(K_1)^{1/2}$ $= (W_2/W_1)(K_2/K_1)^{1/2}$
PEXPOT	1	$(W_2K_2/W_1K_1)^{2/3}$	$(W_2)^{5/3}(K_2)^{2/3} / (W_1)^{5/3}(K_1)^{2/3}$ $= (W_2/W_1)^{5/3}(K_2/K_1)^{2/3}$
DYNPOT	1	W_2K_2/W_1K_1	$(W_2)^2K_2 / (W_1)^2K_1$ $= (W_2/W_1)^2(K_2/K_1)$
LEVPOT	1	$(W_2K_2/W_1K_1)^2$	$(W_2)^3(K_2)^2 / (W_1)^3(K_1)^2$ $= (W_2/W_1)^3(K_2/K_1)^2$

Now consider the special case of the above in which the one type of weapon on side 1 is sufficiently similar (i.e., identical in all relevant aspects) to the one type of weapon on side 2 so that $K_1 = K_2$. Then the table above reduces to the following:

Method	V_1	V_2	$R = W_2V_2/W_1V_1$
APP	1	1	W_2/W_1
APPFUL	1	1	W_2/W_1
PEXPOT	1	$(W_2/W_1)^{2/3}$	$(W_2/W_1)^{5/3}$
DYNPOT	1	W_2/W_1	$(W_2/W_1)^2$
LEVPOT	1	$(W_2/W_1)^2$	$(W_2/W_1)^3$

For example, in this special case if side 2 had three times as many of these essentially identical weapons as side 1 had (i.e., $W_2 = 3W_1$) then the following force ratios would be produced:

Method:	APP	APPVUL	PEXPOT	DYNPOT	LEVPOT
Resulting Force Ratio:	3 to 1	3 to 1	6.24 to 1	9 to 1	27 to 1

An additional requirement that could be imposed on these methods of computing weapons scores is that, in the homogeneous case in which both sides are using essentially identical weapons, if one side has an x to 1 advantage in the number of these identical weapons, then the resulting force ratio is x to 1. If the force ratio is defined to be

$$R = S^2/S^1$$

where

$$S^s = \sum_{n=1}^{N^s} V_i^s W_i^s ,$$

then only APP and APPVUL have this property. However, if a set of force ratios is defined by

$$R_\rho = S_\rho^2 / S_\rho^1$$

where

$$S_\rho^s = \left(\sum_{i=1}^{N^s} v_i^s w_i^s \right)^\rho$$

for $\rho > 0$, then APP and APPVUL have this property for $\rho = 1$, PEXPOT has this property for $\rho = 3/5$, DYNPOT has this property for $\rho = 1/2$, and LEVPOT has this property for $\rho = 1/3$. If force ratios are defined in this manner, then the table below gives the resulting force ratios for the homogeneous case in which the weapons on side 1 are not necessarily identical to the weapons on side 2. Note that, in this homogeneous case, S_ρ^1, S_ρ^2, and R_ρ are identical for APP and DYNPOT. In heterogeneous cases, the values S_ρ^1, S_ρ^2, and P_ρ for DYNPOT with $\rho = 0.5$ need not equal those for APP with $\rho = 1.0$.

Method	ρ	$R_\rho = S_\rho^2 / S_\rho^1$
APP	1	$W_2(K_2)^{1/2} / W_1(K_1)^{1/2}$ $= (W_2/W_1)(K_2/K_1)^{1/2}$
APPVUL	1	$W_2(K_2)^{1/2} / W_1(K_1)^{1/2}$ $= (W_2/W_1)(K_2/K_1)^{1/2}$
PEXPOT	3/5	$W_2(K_2)^{2/5} / W_1(K_1)^{2/5}$ $= (W_2/W_1)(K_2/K_1)^{2/5}$
DYNPOT	1/2	$W_2(K_2)^{1/2} / W_1(K_1)^{1/2}$ $= (W_2/W_1)(K_2/K_1)^{1/2}$
LEVPOT	1/3	$W_2(K_2)^{2/3} / W_1(K_1)^{2/3}$ $= (W_2/W_1)(K_2/K_1)^{2/3}$